Requirements Document

REQ-1: Fix flakey test $ARGUMENTS

REQ-2: Fix issue $ARGUMENTS

REQ-3: Fetch unresolved comments for this branch's PR, then fix them

REQ-4: Repro issue $ARGUMENTS in a failing test

REQ-5: Update `clj-kondo` configs for libraries using  
  
```sh  
clj-kondo --copy-configs --dependencies --lint "$(clojure -Spath -A:dev)"  
```

REQ-6: FROM mcr.microsoft.com/vscode/devcontainers/java:21  
  
# Set up nodesource, install node, yarn, fontconfig for static viz, rlwrap for dev ergonomics  
  
RUN ( curl -fsSL...

REQ-7: .babel\_cache  
docs/\*  
target/\*  
.circleci  
.cpcache  
.devcontainer  
.github  
.husky  
.lsp  
.shadow-cljs  
.github  
.vscode  
hooks/\*  
test/\*  
test\_config/\*  
test\_modules/\*  
test\_resources/\*  
node\_modules  
\*\*metabase.jar  
...

REQ-8: Metabase EditorConfig settings  
http://editorconfig.org/  
use the top most editor config file

REQ-9: frontend/src/cljs  
frontend/src/cljs\_release  
e2e/support/cypress\_sample\_database.js  
e2e/support/cypress\_sample\_instance\_data.js

REQ-10: eslint-disable import/no-commonjs  
eslint-disable no-undef

REQ-11: \*.class  
\*.\*.rej  
\*.iml  
\*.jar  
\*.log  
\*.po~  
\*.sqlite  
\*.trace.db  
\*.trace.db.old  
\*.xcworkspacedata  
.DS\_Store  
.\#\*  
.cpcache/  
.eastwood  
.idea/  
.nrepl-port  
.portal  
.vscode  
/\*.h2.db  
/\*.lock.db  
/\*.mv.db  
/\*.trace...

REQ-12: http://localhost\*  
https://admin.google.com

REQ-13: v22.13.1

REQ-14: frontend/src/cljs  
docs/configuring-metabase/environment-variables.md  
docs/api/\*  
docs/api-documentation.md  
.storybook-sdk/msw-public/mockServiceWorker.js

REQ-15: {  
 "trailingComma": "all"  
}

REQ-16: network-timeout 600000

REQ-17: ##################  
STAGE 1: builder  
##################

REQ-18: GNU AFFERO GENERAL PUBLIC LICENSE  
 Version 3, 19 November 2007  
  
 Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>  
 Everyone is permitted to copy and distribute verbatim...

REQ-19: METABASE APP-EMBED.JS  
SOFTWARE LICENSE AGREEMENT  
  
PLEASE READ THE FOLLOWING TERMS AND CONDITIONS CAREFULLY BEFORE DOWNLOADING, INSTALLING OR USING THE APP-EMBED.JS SOFTWARE OR ANY ACCOMPANYING...

REQ-20: Usage of files in the top-level "/enterprise" directory and subdirectories thereof, and of Metabase Enterprise Edition features, is subject to the Metabase Commercial License...

REQ-21: Source code in this repository is variously licensed under the GNU Affero General Public License (AGPL), or the Metabase Commercial License (https://www.metabase.com/license/commercial).  
  
\* Outside...

REQ-22: analystBase\_qorix\_incubDeployment  
AnalystBase: A Modern Business Intelligence Frontend

REQ-23: 10c975ab6e6f4de9d0f77cec9a3dc793267a8e16  
05b89437a9a43be2d23c25a9ae006ec237240434

REQ-24: # Downloading Jars

REQ-25: !/usr/bin/env node  
eslint-disable no-console  
global process  
This workflow is experimental POC, non-required, and is owned by team  
Embedding. For the time being, unless you think you did a breaking...

REQ-26: Build-drivers scripts

REQ-27: !/usr/bin/env bash

REQ-28: # Build Metabase Tooling

REQ-29: !/bin/bash

REQ-30: !/usr/bin/env node  
This proxy is useful for debugging frontend issues on deployed instances of  
Metabase. You can use the frontend of a local insance in conjunction with them  
backend of a deployed...

REQ-31: FROM eclipse-temurin:21-jre-alpine  
  
ENV FC\_LANG=en-US LC\_CTYPE=en\_US.UTF-8  
  
ARG GIT\_COMMIT\_SHA  
ENV GIT\_COMMIT\_SHA=${GIT\_COMMIT\_SHA}  
  
# dependencies  
RUN apk add -U bash fontconfig curl font-noto...

REQ-32: FROM eclipse-temurin:21-jre-noble as runner  
  
ENV FC\_LANG=en-US LC\_CTYPE=en\_US.UTF-8  
  
ARG GIT\_COMMIT\_SHA  
ENV GIT\_COMMIT\_SHA=${GIT\_COMMIT\_SHA}  
  
# Dependencies  
RUN apt-get update && \  
 apt-get upgrade...

REQ-33: !/usr/bin/env node  
eslint-env node  
eslint-disable import/no-commonjs

REQ-34: !/usr/bin/env node  
eslint-env node  
eslint-disable import/no-commonjs, import/order, no-console

REQ-35: !/usr/bin/env node  
eslint-env node  
eslint-disable import/no-commonjs, import/order, no-console

REQ-36: !/usr/bin/env tsx  
eslint-disable no-console

REQ-37: import {  
 extractVariableDefinitionsFromFileContent,  
 extractVariableUsagesFromFileContent,  
} from "./find-never-defined-css-variables";  
  
describe("extractVariableDefinitionsFromFileContent", () =>...

REQ-38: # i18n info  
### Building the backend pot file

REQ-39: ! /usr/bin/env bash

REQ-40: !/usr/bin/env bash  
This script merges translations from the given branch into the current branch.  
  
It's useful to run this after backporting a translation PR, since that PR may  
remove translations...

REQ-41: ! /usr/bin/env bash

REQ-42: Linter that validates the Liquibase migrations file against a spec. This lets us check and enforce additional  
constraints for the `000\_migrations.yaml` file (e.g. make sure you're not using duplicate...

REQ-43: !/usr/bin/env bash

REQ-44: !/usr/bin/env bb

REQ-45: Release list

REQ-46: ---  
title: Metabase versions  
---  
  
# Metabase versions  
  
Below are links to releases for:  
  
- [Metabase Enterprise Edition](#metabase-enterprise-edition-releases). This edition is used in the...

REQ-47: # Metabase Release Script 3.0  
### Prereqs

REQ-48: ---  
title: {{title}}  
---  
  
# {{title}}  
  
This reference lists all country codes and their corresponding country names used in Metabase's default world map visualizations. The data comes from the...

REQ-49: !node

REQ-50: !/usr/bin/env node

REQ-51: 10c975ab6e6f4de9d0f77cec9a3dc793267a8e16  
05b89437a9a43be2d23c25a9ae006ec237240434

REQ-52: # Render png

REQ-53: ---  
title: Contributing to Metabase  
redirect\_from:  
 - /docs/latest/developers-guide/contributing  
---  
  
# Contributing to Metabase  
  
## Thank you  
  
First off, thanks for your interest in Metabase and...

REQ-54: ---  
title: Metabase documentation  
redirect\_from:  
 - /docs/latest/enterprise-guide  
 - /docs/latest/users-guide  
 - /docs/latest/administration-guide  
 - /docs/latest/operations-guide  
 -...

REQ-55: ---  
title: Basic actions  
---  
  
# Basic actions  
  
Basic actions are "implicit" [actions](./introduction.md) that do things that people typically want to do when interacting with a database: Create,...

REQ-56: ---  
title: Custom actions  
---  
  
# Custom actions  
  
Write SQL to update records in your databases.  
  
![Custom action](./images/custom-action.png)  
  
## Creating a custom action  
  
> You must be in a group...

REQ-57: ---  
title: Introduction to actions  
---  
  
# Introduction to actions  
  
> For now, actions are only available for PostgreSQL and MySQL.  
  
![Example action](./images/example-action.png)  
  
## What are...

REQ-58: ---  
title: Actions overview  
redirect\_from:  
 - /docs/latest/actions  
---  
  
# Actions overview  
  
![An action updating a plan on a dashboard](./images/dashboard-action.gif)  
  
Actions let you write...

REQ-59: ---  
title: "Accounts and billing"  
version: latest  
has\_magic\_breadcrumbs: true  
show\_category\_breadcrumb: true  
show\_title\_breadcrumb: true  
category: "Cloud"  
layout: new-docs  
---  
  
# Accounts and...

REQ-60: ---  
title: "Changing which region your Metabase is hosted in"  
version: latest  
has\_magic\_breadcrumbs: true  
show\_category\_breadcrumb: true  
show\_title\_breadcrumb: true  
category: "Cloud"  
layout:...

REQ-61: ---  
title: "Metabase Cloud versus self-hosting"  
version: latest  
has\_magic\_breadcrumbs: true  
show\_category\_breadcrumb: true  
show\_title\_breadcrumb: true  
category: "Cloud"  
layout: new-docs  
---  
  
#...

REQ-62: ---  
title: "Changing your domain name"  
version: latest  
has\_magic\_breadcrumbs: true  
show\_category\_breadcrumb: true  
show\_title\_breadcrumb: true  
category: "Cloud"  
layout: new-docs  
redirect\_from:  
 -...

REQ-63: ---  
title: Sync Google Sheets with Metabase  
version: latest  
has\_magic\_breadcrumbs: true  
show\_category\_breadcrumb: true  
show\_title\_breadcrumb: true  
category: "Cloud"  
layout: new-docs  
---  
  
# Sync...

REQ-64: ---  
title: "How Metabase billing works"  
version: latest  
has\_magic\_breadcrumbs: true  
show\_category\_breadcrumb: true  
show\_title\_breadcrumb: true  
category: "Cloud"  
layout: new-docs  
redirect\_from:  
 -...

REQ-65: ---  
title: "IP addresses to whitelist"  
version: latest  
has\_magic\_breadcrumbs: true  
show\_category\_breadcrumb: true  
show\_title\_breadcrumb: true  
category: "Cloud"  
layout: new-docs  
redirect\_from:  
 -...

REQ-66: ---  
title: "Limitations of Metabase Cloud"  
version: latest  
has\_magic\_breadcrumbs: true  
show\_category\_breadcrumb: true  
show\_title\_breadcrumb: true  
category: "Cloud"  
layout: new-docs  
redirect\_from:  
 -...

REQ-67: ---  
title: "Migrate from Metabase Cloud to a self-hosted Metabase"  
version: latest  
has\_magic\_breadcrumbs: true  
show\_category\_breadcrumb: true  
show\_title\_breadcrumb: true  
category: "Cloud"  
layout:...

REQ-68: ---  
title: "Migrate to Metabase Cloud - Metabase 49 or lower"  
version: latest  
has\_magic\_breadcrumbs: true  
show\_category\_breadcrumb: true  
show\_title\_breadcrumb: true  
category: "Cloud"  
layout:...

REQ-69: ---  
title: "Migrate to Metabase Cloud"  
version: latest  
has\_magic\_breadcrumbs: true  
show\_category\_breadcrumb: true  
show\_title\_breadcrumb: true  
category: "Cloud"  
layout: new-docs  
section:...

REQ-70: ---  
title: "Migrating from Heroku to Metabase Cloud"  
version: latest  
has\_magic\_breadcrumbs: true  
show\_category\_breadcrumb: true  
show\_title\_breadcrumb: true  
category: "Cloud"  
layout:...

REQ-71: ---  
title: Metabase Cloud  
version: latest  
has\_magic\_breadcrumbs: true  
show\_category\_breadcrumb: true  
show\_title\_breadcrumb: false  
category: "Cloud"  
redirect\_from:  
 - /cloud/docs  
---  
  
# Metabase...

REQ-72: ---  
title: Metabase Cloud Storage  
version: latest  
has\_magic\_breadcrumbs: true  
show\_category\_breadcrumb: true  
show\_title\_breadcrumb: true  
category: "Cloud"  
layout: new-docs  
---  
  
# Metabase Cloud...

REQ-73: ---  
title: Appearance  
redirect\_from:  
 - /docs/latest/administration-guide/whitelabeling  
 - /docs/latest/enterprise-guide/whitelabeling  
 - /docs/latest/enterprise-guide/customize-embeds  
 -...

REQ-74: ---  
title: Caching query results  
redirect\_from:  
 - /docs/latest/administration-guide/14-caching  
 - /docs/latest/enterprise-guide/cache  
---  
  
# Caching query results  
  
If your question results don't...

REQ-75: ---  
title: "Configuration file"  
---  
  
# Configuration file  
  
{% include plans-blockquote.html feature="Loading from a configuration file" self-hosted-only="true" %}  
  
On self-hosted...

REQ-76: ---  
title: "Metabase config file template"  
---  
  
# Metabase config file template  
  
You can generate the following docs by changing into the top-level Metabase directory and running:  
  
```  
clojure...

REQ-77: ---  
title: Custom maps  
redirect\_from:  
 - /docs/latest/administration-guide/20-custom-maps  
---  
  
# Custom maps  
  
By default, Metabase uses OpenStreetMaps for map visualizations, but there are a few...

REQ-78: ---  
title: Customizing the Metabase Jetty webserver  
redirect\_from:  
 - /docs/latest/operations-guide/customizing-jetty-webserver  
---  
  
# Customizing the Metabase Jetty webserver  
  
In most cases there...

REQ-79: ---  
title: Set up email  
summary: Learn how to set up email in Metabase to enable dashboard subscriptions and alerts.  
redirect\_from:  
 - /docs/latest/administration-guide/02-setting-up-email  
---  
  
#...

REQ-80: ---  
title: Environment variables  
redirect\_from:  
 - /docs/latest/operations-guide/environment-variables  
---  
  
# Environment variables  
  
\_This documentation was generated from source by...

REQ-81: ---  
title: Fonts  
redirect\_from:  
 - /docs/latest/enterprise-guide/fonts  
 - /docs/latest/embedding/fonts  
---  
  
# Fonts  
  
{% include plans-blockquote.html feature="Customizable font" %}  
  
On Pro and...

REQ-82: ---  
title: Languages and localization  
redirect\_from:  
 - /docs/latest/administration-guide/localization  
---  
  
# Languages and localization  
  
Admins can update the localization settings for the...

REQ-83: ---  
title: Metabase logs  
redirect\_from:  
 - /docs/latest/operations-guide/log-configuration  
summary: Configure how much information Metabase displays in its logs.  
---  
  
# Metabase logs  
  
Metabase logs...

REQ-84: ---  
title: Setting up Metabase  
redirect\_from:  
 - /docs/latest/setting-up-metabase  
---  
  
# Setting up Metabase  
  
This guide will help you set up Metabase once you've gotten it installed. If you haven't...

REQ-85: ---  
title: General settings  
redirect\_from:  
 - /docs/latest/administration-guide/08-configuration-settings  
---  
  
# General settings  
  
This section contains settings for your whole instance, like its...

REQ-86: ---  
title: Set up Slack  
redirect\_from:  
 - /docs/latest/administration-guide/09-setting-up-slack  
---  
  
# Set up Slack  
  
If you want to have your [dashboard...

REQ-87: ---  
title: "Configuration overview"  
redirect\_from:  
 - /docs/latest/configuring-metabase  
---  
  
# Configuration overview  
  
## [Setting up Metabase](./setting-up-metabase.md)  
  
A walkthrough of when you...

REQ-88: ---  
title: Timezones  
redirect\_from:  
 - /docs/latest/operations-guide/handling-timezones  
---  
  
# Timezones  
  
Metabase does its best to ensure proper and accurate reporting in whatever timezone you...

REQ-89: ---  
title: Webhooks  
---  
  
# Webhooks  
  
Admins can set up webhooks so that people can send [alerts](../questions/alerts.md) to a particular URL. Which means you can set up an alert to send the results...

REQ-90: ---  
title: Actions on dashboards  
---  
  
# Actions on dashboards  
  
![Dashboard with filter, action button, and detail card view](./images/dashboard-filter-action.png)  
  
To put...

REQ-91: ---  
title: Dashboard filters  
summary: Make your dashboards do more with filters and parameters. Instead of creating a bunch of similar dashboards, just add widgets to change...

REQ-92: ---  
title: Dashboard interactivity  
redirect\_from:  
 - /docs/latest/users-guide/interactive-dashboards  
---  
  
# Dashboard interactivity  
  
You can customize what happens when people click on questions in...

REQ-93: ---  
title: Introduction to dashboards  
redirect\_from:  
 - /docs/latest/users-guide/07-dashboards  
---  
  
# Introduction to dashboards  
  
![Interactive dashboard](./images/interactive-dashboard.png)  
  
##...

REQ-94: ---  
title: Linked filters  
---  
  
# Linked filters  
  
You can \*\*link filters\*\* on a dashboard so that a child filter limits its values based on the value(s) applied by a parent filter.  
  
For example, let's...

REQ-95: ---  
title: Charts with multiple series  
redirect\_from:  
 - /docs/latest/users-guide/09-multi-series-charting  
---  
  
# Charts with multiple series  
  
One of the best ways to add context and clarity when...

REQ-96: ---  
title: Dashboards overview  
redirect\_from:  
 - /docs/latest/dashboards  
---  
  
# Dashboards overview  
  
![Example dashboard](./images/dashboard.png)  
  
## [Introduction to...

REQ-97: ---  
title: Dashboard subscriptions  
redirect\_from:  
 - /docs/latest/users-guide/dashboard-subscriptions  
 - /docs/latest/enterprise-guide/dashboards-subscriptions  
---  
  
# Dashboard...

REQ-98: ---  
title: Formatting defaults  
summary: Configure how dates, numbers, currencies, and text display in Metabase at global, field, and question levels.  
redirect\_from:  
 -...

REQ-99: ---  
title: Working with JSON  
summary: Learn how to unfold JSON columns into separate fields that you can filter on in the query builder.  
---  
  
# Working with JSON  
  
## Filtering JSON  
  
In the [query...

REQ-100: ---  
title: Legacy metrics  
---  
  
# Legacy metrics  
  
Metabase upgraded [metrics](./metrics.md) starting in version 51.  
  
If you're upgrading from Metabase 50 or earlier, all of your existing metrics will...

REQ-101: ---  
title: "Table metadata admin settings"  
redirect\_from:  
 - /docs/latest/administration-guide/03-metadata-editing  
---  
  
# Table metadata admin settings  
  
\_Admin settings > Table metadata\_  
  
![Table...

REQ-102: ---  
title: Metrics  
redirect\_from:  
 - /docs/latest/administration-guide/07-segments-and-metrics  
 - /docs/latest/data-modeling/segments-and-metrics  
---  
  
# Metrics  
  
Create metrics to define the...

REQ-103: ---  
title: Model persistence  
---  
  
# Model persistence  
  
> Currently available for PostgreSQL, MySQL, and Redshift.  
  
Metabase can persist the results of your models so that your models (and the...

REQ-104: ---  
title: "Models"  
redirect\_from:  
 - /docs/latest/users-guide/models  
---  
  
# Models  
  
Models are a fundamental building block in Metabase. Models curate data from another table or tables from the...

REQ-105: ---  
title: "Segments"  
---  
  
# Segments  
  
Metabase allows admins to create segments so people can quickly and easily reference them in the query builder.  
  
To manage segments:  
  
1. Click the \*\*gear\*\* icon...

REQ-106: ---  
title: "Data types and semantic types"  
redirect\_from:  
 - /docs/latest/users-guide/field-types  
 - /docs/latest/data-modeling/field-types  
summary: "Metabase uses both data and semantic types to...

REQ-107: ---  
title: "Data modeling overview"  
redirect\_from:  
 - /docs/latest/data-modeling  
---  
  
# Data modeling overview  
  
Metabase provides tools for organizing your data and making it easier for people to...

REQ-108: ---  
title: Adding and managing databases  
redirect\_from:  
 - /docs/latest/administration-guide/01-managing-databases  
 - /docs/latest/databases/connections/sql-server  
 -...

REQ-109: ---  
title: Amazon Athena  
---  
  
# Amazon Athena  
  
To add a database connection, click on the \*\*gear\*\* icon in the top right, and navigate to \*\*Admin settings\*\* > \*\*Databases\*\* > \*\*Add a database\*\*.  
  
##...

REQ-110: ---  
title: "Connecting to AWS's Relational Database Service (RDS)"  
redirect\_from:  
 - /docs/latest/administration-guide/databases/aws-rds  
---  
  
# Connecting to AWS's Relational Database Service...

REQ-111: ---  
title: Google BigQuery  
redirect\_from:  
 - /docs/latest/administration-guide/databases/bigquery  
---  
  
# Google BigQuery  
  
To add a database connection, click on the \*\*gear\*\* icon in the top right,...

REQ-112: ---  
title: ClickHouse  
description: Learn how to connect Metabase to your ClickHouse database, including connection settings, database selection, and SSL configuration.  
---  
  
# ClickHouse  
  
To add a...

REQ-113: ---  
title: Databricks  
---  
  
# Databricks  
  
To add a database connection, click on the \*\*gear\*\* icon in the top right, and navigate to \*\*Admin settings\*\* > \*\*Databases\*\* > \*\*Add a database\*\*. Then...

REQ-114: ---  
title: Druid  
---  
  
# Druid  
  
> Prefer the Druid JDBC connection.  
  
To add a database connection, click on the \*\*gear\*\* icon in the top right, and navigate to \*\*Admin settings\*\* > \*\*Databases\*\* >...

REQ-115: ---  
title: MariaDB  
---  
  
# MariaDB  
  
> This page covers connecting to MariaDB as a \_data warehouse\_. For using MariaDB as Metabase's \_application database\_, see [Configuring the Metabase application...

REQ-116: ---  
title: MongoDB  
redirect\_from:  
 - /docs/latest/administration-guide/databases/mongodb  
---  
  
# MongoDB  
  
To add a database connection, click on the \*\*gear\*\* icon in the top right, and navigate to...

REQ-117: ---  
title: MySQL  
redirect\_from:  
 - /docs/latest/administration-guide/databases/mysql  
---  
  
# MySQL  
  
> This page covers connecting to MySQL as a \_data warehouse\_. For using MySQL as Metabase's...

REQ-118: ---  
title: Oracle  
redirect\_from:  
 - /docs/latest/administration-guide/databases/oracle  
---  
  
# Oracle  
  
To add a database connection, click on the \*\*gear\*\* icon in the top right, and navigate to...

REQ-119: ---  
title: PostgreSQL  
redirect\_from:  
 - /docs/latest/administration-guide/databases/postgresql  
---  
  
# PostgreSQL  
  
> This page covers connecting to PostgreSQL as a \_data warehouse\_. For using...

REQ-120: ---  
title: Presto  
---  
  
# Presto  
  
To add a database connection, click on the \*\*gear\*\* icon in the top right, and navigate to \*\*Admin settings\*\* > \*\*Databases\*\* > \*\*Add a database\*\*.  
  
Fill out the...

REQ-121: ---  
title: Amazon Redshift  
description: Connect Metabase to Amazon Redshift data warehouses to run queries and create dashboards.  
redirect\_from:  
 -...

REQ-122: ---  
title: Snowflake  
redirect\_from:  
 - /docs/latest/administration-guide/databases/snowflake  
---  
  
# Snowflake  
  
To add a database connection, click on the \*\*gear\*\* icon in the top right, and navigate...

REQ-123: ---  
title: SparkSQL  
---  
  
# SparkSQL  
  
To add a database connection, click on the \*\*gear\*\* icon in the top right, and navigate to \*\*Admin settings\*\* > \*\*Databases\*\* > \*\*Add a database\*\*.  
  
##...

REQ-124: ---  
title: SQL Server  
---  
  
# SQL Server  
  
To add a database connection, click on the \*\*gear\*\* icon in the top right, and navigate to \*\*Admin settings\*\* > \*\*Databases\*\* > \*\*Add a database\*\*.  
  
##...

REQ-125: ---  
title: SQLite  
---  
  
# SQLite  
  
> SQLite isn't available for [Metabase Cloud](https://www.metabase.com/cloud/).  
  
To add a database connection, click on the \*\*gear\*\* icon in the top right, and...

REQ-126: ---  
title: Starburst  
description: Learn how to connect Metabase to your Starburst or Trino database, including connection settings, SSL configuration, and database sync options.  
---  
  
# Starburst  
  
>...

REQ-127: ---  
title: Working with Vertica in Metabase  
redirect\_from:  
 - /docs/latest/administration-guide/databases/vertica  
---  
  
# Working with Vertica in Metabase  
  
Starting in v0.20.0, Metabase provides a...

REQ-128: ---  
title: Danger zone  
description: The Danger Zone section of database connections is where you can discard field values or remove database connections and all of their related content.  
---  
  
#...

REQ-129: ---  
title: Encrypting your database connection  
redirect\_from:  
 - /docs/latest/operations-guide/encrypting-database-details-at-rest  
---  
  
# Encrypting your database connection  
  
Metabase stores...

REQ-130: ---  
title: SSH tunneling  
redirect\_from:  
 - /docs/latest/administration-guide/ssh-tunnel-for-database-connections  
---  
  
# SSH tunneling  
  
Metabase can connect to some databases by first establishing a...

REQ-131: ---  
title: SSL certificate  
redirect\_from:  
 - /docs/latest/administration-guide/secure-database-connections-with-ssl-certificates  
---  
  
# SSL certificate  
  
If you'd like to connect your Metabase Cloud...

REQ-132: ---  
title: Databases overview  
redirect\_from:  
 - /docs/latest/databases  
---  
  
# Databases overview  
  
## [Adding and managing databases](./connecting.md)  
  
Connect to and manage your databases.  
  
##...

REQ-133: ---  
title: Syncing and scanning databases  
summary: Learn how Metabase stays in sync with your database by running periodic queries to update metadata, sample field values, and compute stats.  
---  
  
#...

REQ-134: ---  
title: Setting up data uploads  
---  
  
# Setting up data uploads  
  
This page covers how admins can set up data uploads so people can upload CSVs to your Metabase. For \_how\_ to upload data once this...

REQ-135: ---  
title: Database users, roles, and privileges  
---  
  
# Database users, roles, and privileges  
  
We recommend creating a `metabase` database user with the following database roles:  
  
- [`analytics` for...

REQ-136: ---  
title: API changelog  
---  
  
# Breaking changes to the API interface  
  
## Metabase 0.55.0  
  
- `POST /api/card/from-csv` has been renamed to `POST /api/upload/csv`.  
  
- `GET /api/util/stats` has been...

REQ-137: ---  
title: Building Metabase  
---  
  
# Building Metabase  
  
This doc will show you how you can build and run Metabase on your own computer so you can play around with it or test features in development....

REQ-138: ---  
title: Working with Clojure  
---  
  
# Working with Clojure  
  
Check out [Clojure for the Brave and True](https://www.braveclojure.com/clojure-for-the-brave-and-true/). It's free online.  
  
If you don't...

REQ-139: ---  
title: "Code reviews"  
redirect\_from:  
 - /docs/latest/code-reviews  
---  
  
# Code reviews  
  
The overall goal of a code review is to serve as a safety net for other people on our team and help them...

REQ-140: ---  
title: Community drivers  
redirect\_from:  
 - /docs/latest/developers-guide-drivers  
 - /docs/latest/developers-guide/partner-and-community-drivers  
---  
  
# Community drivers  
  
> Community drivers are...

REQ-141: ---  
title: How to run a development branch of Metabase using Docker  
---  
  
# How to run a development branch of Metabase using Docker  
  
If you want to run a branch of Metabase that's currently in...

REQ-142: ---  
title: Development environment  
---  
  
# Development environment  
  
The Metabase application has two basic components:  
  
1. A backend written in Clojure which contains a REST API as well as all the...

REQ-143: ---  
title: Developing Metabase documentation  
---  
  
# Developing Metabase documentation  
  
Notes on writing docs for Metabase.  
  
## Linting markdown links  
  
You can check for broken links in the...

REQ-144: ---  
title: Driver interface changelog  
---  
  
# Driver Interface Changelog  
  
## Metabase 0.56.0  
  
- Add the testing multi-method `tx/track-dataset` for shared cloud dbs to track loaded datasets for more...

REQ-145: ---  
title: Database driver basics  
---  
  
# Database driver basics  
  
A Metabase driver:  
  
- \*\*Provides Metabase with basic information for the database\*\* such as the database's capabilities, connection...

REQ-146: ---  
title: Submitting a PR for a new driver  
---  
  
# Submitting a PR for a new driver  
  
If you want to submit a PR to add a driver plugin to the [Metabase repo](https://github.com/metabase/metabase) (as...

REQ-147: ---  
title: Implementing multimethods for your driver  
---  
  
# Implementing multimethods for your driver  
  
Implementing multimethods lets you take advantage of Metabase's existing driver code by...

REQ-148: ---  
title: Plugin manifests  
---  
  
# Plugin manifests  
  
Metabase plugin JARs contain a \_plugin manifest\_ -- a top-level file named `metabase-plugin.yaml`. When Metabase launches, it iterates over every...

REQ-149: ---  
title: Guide to writing a Metabase driver  
---  
  
# Guide to writing a Metabase driver  
  
So here's the scenario: you love Metabase. It's changed your life. But you have some data in a Visual Fox Pro...

REQ-150: ---  
title: End-to-end tests with Cypress  
---  
  
# End-to-end tests with Cypress  
  
Metabase uses Cypress for "end-to-end testing", that is, tests that are executed against the application as a whole,...

REQ-151: ---  
title: Developing Metabase with Emacs  
---  
  
# Developing Metabase with Emacs  
  
`.dir-locals.el` contains some Emacs Lisp that tells `clojure-mode` how to indent Metabase macros and which arguments...

REQ-152: ---  
title: Frontend  
---  
  
# Frontend  
  
## Entity Loaders  
  
If you're developing a new feature or just generally need to get at some of the application data on the frontend, Entity Loaders are going to...

REQ-153: ---  
title: Internationalization  
---  
  
# Internationalization  
  
We are an application with lots of users all over the world. To help them use Metabase in their own language, we mark all of our strings...

REQ-154: MAGE - Development Automation

REQ-155: ---  
title: MBQL Library changelog  
---  
  
# MBQL Library Changelog  
  
Changes made to the library API for manipulating MBQL queries, found in `metabase.lib.js`. The latest API documentation  
can be found...

REQ-156: Security Token Scanner

REQ-157: ---  
title: "Developer Guide"  
redirect\_from:  
 - /docs/latest/developers-guide  
---  
  
# Developer Guide  
  
This guide contains detailed information on how to work on Metabase codebase.  
  
## Contributing  
  
-...

REQ-158: ---  
title: Metabase release versioning  
---  
  
# Metabase release versioning  
  
We follow our own flavor of the [semantic versioning guidelines](https://semver.org/) in order to distinguish the...

REQ-159: ---  
title: Developing with Visual Studio Code  
---  
  
# Developing with Visual Studio Code  
  
## Debugging  
  
First, install the following extension:  
  
- [Debugger for...

REQ-160: ---  
title: Visual Tests  
---  
  
# Visual Tests  
  
We use [Loki](https://loki.js.org/) with Storybook for visual tests. Loki captures snapshots from selected stories and creates PNG references in...

REQ-161: ---  
title: Embedded Analytics JS  
summary: Getting started with Embedded Analytics JS for embedding Metabase entities into external applications  
---  
  
# Embedded Analytics JS  
  
Embedded analytics JS...

REQ-162: ---  
title: "Interactive embedding quickstart"  
---  
  
# Interactive embedding quickstart  
  
You'll embed the full Metabase application in your app. Once logged in, people can view a Metabase dashboard in...

REQ-163: ---  
title: Interactive embedding  
redirect\_from:  
 - /docs/latest/enterprise-guide/full-app-embedding  
 - /docs/latest/embedding/full-app-embedding  
---  
  
# Interactive embedding  
  
{% include...

REQ-164: ---  
title: Interactive embedding UI components  
description: Customize the UI components in your interactive Metabase embed by adding parameters to the embedding URL.  
---  
  
# Interactive embedding UI...

REQ-165: ---  
title: Embedding introduction  
redirect\_from:  
 - /docs/latest/administration-guide/13-embedding  
---  
  
# Embedding introduction  
  
You can embed Metabase tables, charts, and dashboards-even...

REQ-166: ---  
title: Public sharing  
redirect\_from:  
 - /docs/latest/administration-guide/12-public-links  
 - /docs/latest/embedding/12-public-links  
---  
  
# Public sharing  
  
> Only admins can create public links...

REQ-167: ---  
title: "Embedded analytics SDK - appearance"  
---  
  
# Embedded analytics SDK - appearance  
  
{% include plans-blockquote.html feature="Embedded analytics SDK" sdk=true %}  
  
You can style your embedded...

REQ-168: ---  
title: Embedded analytics SDK - authentication  
---  
  
# Embedded analytics SDK - authentication  
  
{% include plans-blockquote.html feature="Embedded analytics SDK" sdk=true %}  
  
For using the SDK in...

REQ-169: ---  
title: Embedded analytics SDK - collections  
---  
  
# Embedded analytics SDK - collections  
  
{% include plans-blockquote.html feature="Embedded analytics SDK" sdk=true %}  
  
## Embedding a collection...

REQ-170: ---  
title: Embedded analytics SDK - config  
---  
  
# Embedded analytics SDK - config  
  
{% include plans-blockquote.html feature="Embedded analytics SDK" sdk=true %}  
  
## Passing a configuration object to...

REQ-171: ---  
title: "Embedded analytics SDK - dashboards"  
---  
  
# Embedded analytics SDK - dashboards  
  
{% include plans-blockquote.html feature="Embedded analytics SDK" sdk=true %}  
  
You can embed an...

REQ-172: ---  
title: Embedded analytics SDK  
redirect\_from:  
 - /docs/latest/embedding/sdk  
---  
  
# Embedded analytics SDK  
  
{% include plans-blockquote.html feature="Embedded analytics SDK" sdk=true %}  
  
With the...

REQ-173: ---  
title: Embedded analytics SDK - Using the SDK with Next.js  
---  
  
# Embedded analytics SDK - Using the SDK with Next.js  
  
{% include plans-blockquote.html feature="Embedded analytics SDK" sdk=true...

REQ-174: ---  
title: Embedded analytics SDK - plugins  
---  
  
# Embedded analytics SDK - plugins  
  
{% include plans-blockquote.html feature="Embedded analytics SDK" sdk=true %}  
  
The Metabase Embedded analytics SDK...

REQ-175: ---  
title: "Embedded analytics SDK - questions"  
description: How to embed charts in your app with the Embedded analytics SDK.  
---  
  
# Embedded analytics SDK - questions  
  
{% include...

REQ-176: ---  
title: Embedded analytics SDK - CLI quickstart  
---  
  
# Embedded analytics SDK - CLI quickstart  
  
{% include plans-blockquote.html feature="Embedded analytics SDK" sdk=true...

REQ-177: ---  
title: Embedded analytics SDK - quickstart with sample app  
---  
  
# Embedded analytics SDK - quickstart with sample app  
  
{% include plans-blockquote.html feature="Embedded analytics SDK" sdk=true...

REQ-178: ---  
title: Embedded analytics SDK - quickstart  
description: "This guide walks you through how to set up the Embedded analytics SDK in your application with your Metabase."  
---  
  
# Embedded analytics...

REQ-179: import { defineMetabaseTheme } from "@metabase/embedding-sdk-react";  
  
const theme = defineMetabaseTheme({  
 // Specify a font to use from the set of fonts supported by Metabase.  
 // You can set the...

REQ-180: import cors from "cors";  
import express from "express";  
import session from "express-session";  
  
const PORT = 3000;  
const SESSION\_SECRET = "SECRET";  
  
const app = express();  
const metabaseAuthHandler =...

REQ-181: import express from "express";  
import jwt from "jsonwebtoken";  
  
// Replace this with your Metabase URL  
const METABASE\_INSTANCE\_URL = "YOUR\_METABASE\_URL\_HERE";  
// Replace this with the JWT signing...

REQ-182: import express from "express";  
import cors from "cors";  
import session from "express-session";  
import jwt from "jsonwebtoken";  
import fetch from "node-fetch";  
  
// Replace this with your Metabase...

REQ-183: [<snippet imports>]

REQ-184: Required to properly resolve the `import { StaticQuestion } from "@/metabase";` in `manual-wrapping-usage.tsx` snippet

REQ-185: [<snippet imports>]

REQ-186: ---  
title: Upgrading Metabase and the Embedded analytics SDK  
summary: How to upgrade your Metabase and Embedded analytics SDK versions, test the changes, and check for breaking changes that might...

REQ-187: ---  
title: Embedded analytics SDK - versions  
---  
  
# Embedded analytics SDK - versions  
  
{% include plans-blockquote.html feature="Embedded analytics SDK" sdk=true %}  
  
The SDK stable version tracks...

REQ-188: ---  
title: Securing embedded Metabase  
summary: How to hide and protect sensitive data in different types of embeds.  
---  
  
# Securing embedded Metabase  
  
## Securing embeds with authentication and...

REQ-189: import express from "express";  
// [<snippet jsonwebtoken-import>]  
import jwt from "jsonwebtoken";  
// [<endsnippet jsonwebtoken-import>]  
  
const app = express();  
  
const METABASE\_INSTANCE\_URL =...

REQ-190: ---  
title: Embedding overview  
redirect\_from:  
 - /docs/latest/embedding  
---  
  
# Embedding overview  
  
## [Introduction](./introduction.md)  
  
What is embedding, and how does it work?  
  
## [Interactive...

REQ-191: ---  
title: Parameters for static embeds  
summary: Using parameters to customize embedded static questions and dashboards.  
redirect\_from:  
 - /docs/latest/embedding/signed-embedding-parameters  
---  
  
#...

REQ-192: ---  
title: Static embedding  
redirect\_from:  
 - /docs/latest/embedding/signed-embedding  
---  
  
# Static embedding  
  
Also known as: standalone embedding, or signed embedding.  
  
{% include...

REQ-193: ---  
title: Translate embedded dashboards and questions  
summary: Upload a translation dictionary to translate questions and dashboards into different languages. Only available for static...

REQ-194: ---  
title: Collections  
redirect\_from:  
 - /docs/latest/users-guide/collections  
---  
  
# Collections  
  
Collections are the main way to organize [questions](../questions/introduction.md),...

REQ-195: ---  
title: Content verification  
---  
  
# Content verification  
  
{% include plans-blockquote.html feature="Content verification" %}  
  
![Verified icon](./images/verified-icon.png)  
  
Admins can verify items...

REQ-196: ---  
title: Data reference  
redirect\_from:  
 - /docs/latest/users-guide/12-data-model-reference  
 - /docs/latest/questions/native-editor/data-model-reference  
---  
  
# Data reference  
  
You can open the...

REQ-197: ---  
title: Delete and restore  
---  
  
# Delete and restore  
  
Sometimes your questions, dashboards, models, or collections outlive their usefulness. You can send outdated items to \*\*Trash\*\*.  
  
![Move to...

REQ-198: ---  
title: Events and timelines  
redirect\_from:  
 - /docs/latest/users-guide/events-and-timelines  
---  
  
# Events and timelines  
  
A lot of discussions around data have a moment when someone asks a...

REQ-199: ---  
title: Basic exploration  
redirect\_from:  
 - /docs/latest/users-guide/03-basic-exploration  
---  
  
# Basic exploration  
  
## See what your teammates have made  
  
As long as you're not the very first user...

REQ-200: ---  
title: History  
---  
  
# History  
  
For questions, dashboards, and models, Metabase keeps a version history for the previous fifteen versions of that item. You can view changes, and revert to previous...

REQ-201: ---  
title: Keyboard shortcuts  
summary: A handy list of keyboard shortcuts to help you zip around Metabase.  
---  
  
# Keyboard shortcuts  
  
A handy list of keyboard shortcuts to help you zip around...

REQ-202: ---  
title: "Organization overview"  
redirect\_from:  
 - /docs/latest/exploration-and-organization  
---  
  
# Organization overview  
  
Tools for finding things and keeping your Metabase organized.  
  
## [Basic...

REQ-203: ---  
title: Uploading data  
---  
  
# Uploading data  
  
If an admin has [set up uploads](../databases/uploads.md), you can upload CSV data by clicking on the \*\*Upload\*\* icon in the top right of the...

REQ-204: ---  
title: X-rays  
redirect\_from:  
 - /docs/latest/users-guide/14-x-rays  
---  
  
# X-rays  
  
X-rays are a way to get automatic insights and explorations of your data.  
  
## Get automatic insights when...

REQ-205: ---  
title: Accessibility in Metabase  
redirect\_from:  
 - /docs/latest/accessibility  
 - /docs/latest/people-and-groups/accessibility  
---  
  
# Accessibility in Metabase  
  
While we're working to make a...

REQ-206: ---  
title: Activating your Metabase commercial license  
redirect\_from:  
 - /docs/latest/enterprise-guide/activating-the-enterprise-edition  
 -...

REQ-207: ---  
title: Backing up Metabase  
redirect\_from:  
 - /docs/latest/operations-guide/backing-up-metabase-application-data  
---  
  
# Backing up Metabase  
  
Avoid losing your application data (all of your...

REQ-208: ---  
title: Metabase CLI  
---  
  
# Metabase CLI  
  
Metabase ships with some handy CLI commands. To view a list of commands, run the Metabase jar followed by `help`.  
  
```  
java --add-opens...

REQ-209: ---  
title: Configuring the Metabase application database  
redirect\_from:  
 - /docs/latest/operations-guide/configuring-application-database  
---  
  
# Configuring the Metabase application database  
  
The...

REQ-210: ---  
title: "Creating an RDS database on AWS"  
redirect\_from:  
 - /docs/latest/operations-guide/creating-RDS-database-on-AWS  
---  
  
# Creating an RDS database on AWS  
  
If you want to move from using...

REQ-211: ---  
title: Development instances  
summary: Create development instances of Metabase for testing without paying per user. Perfect for trying out changes before pushing them to production.  
---  
  
#...

REQ-212: ---  
title: About the anonymous usage data we collect  
redirect\_from:  
 - /docs/latest/information-collection  
---  
  
# About the anonymous usage data we collect  
  
If you're self-hosting Metabase and...

REQ-213: ---  
title: Installing Metabase  
redirect\_from:  
 - /docs/latest/operations-guide/installing-metabase  
---  
  
# Installing Metabase  
  
Metabase is built and packaged as a Java JAR file and can be run...

REQ-214: ---  
title: Migrating to a production application database  
redirect\_from:  
 - /docs/latest/operations-guide/migrating-from-h2  
 - /docs/latest/operations-guide/running-migrations-manually  
---  
  
#...

REQ-215: ---  
title: Monitoring your Metabase  
redirect\_from:  
 - /docs/latest/operations-guide/jmx-monitoring  
 - /docs/latest/operations-guide/enable-jmx  
---  
  
# Monitoring Your Metabase  
  
Diagnosing...

REQ-216: ---  
title: Observability with Prometheus  
---  
  
# Observability with Prometheus  
  
You can export metrics in [Prometheus](https://prometheus.io/) format from your Metabase.  
  
## Running Metabase and...

REQ-217: ---  
title: Privacy  
redirect\_from:  
 - /docs/latest/privacy  
---  
  
# Privacy  
  
## Do you need a Data Processing Agreement with Metabase to comply with GDPR?  
  
### Self-hosted  
  
If you self-host your...

REQ-218: ---  
title: Running Metabase on Microsoft Azure  
redirect\_from:  
 - /docs/latest/operations-guide/running-metabase-on-azure  
---  
  
# Running Metabase on Microsoft Azure  
  
This guide covers the basics for...

REQ-219: ---  
title: Running Metabase on Debian as a service with nginx  
redirect\_from:  
 - /docs/latest/operations-guide/running-metabase-on-debian  
---  
  
# Running Metabase on Debian as a service with...

REQ-220: ---  
title: Running Metabase on Docker  
redirect\_from:  
 - /docs/latest/operations-guide/running-metabase-on-docker  
---  
  
# Running Metabase on Docker  
  
> To get fast, reliable, and secure deployment...

REQ-221: ---  
title: Running Metabase on AWS Elastic Beanstalk  
redirect\_from:  
 - /docs/latest/operations-guide/running-metabase-on-elastic-beanstalk  
 -...

REQ-222: ---  
title: Running Metabase on Podman  
redirect\_from:  
 - /docs/latest/operations-guide/running-metabase-on-podman  
---  
  
# Running Metabase on Podman  
  
Our official Metabase Docker image is compatible...

REQ-223: ---  
title: Running the Metabase JAR file  
redirect\_from:  
 - /docs/latest/operations-guide/running-the-metabase-jar-file  
 - /docs/installation-and-operation/java-versions  
---  
  
# Running the Metabase...

REQ-224: ---  
title: "Serialization"  
summary: How to export and import Metabase content between instances using serialization. Useful for version control, staging environments, and duplicating...

REQ-225: ---  
title: "Installation and operation overview"  
---  
  
# Installation and operation overview  
  
The birth, care, and feeding of your Metabase.  
  
## [Installing...

REQ-226: ---  
title: Supported browsers  
redirect\_from:  
 - /docs/latest/administration-guide/supported-browsers  
---  
  
# Supported browsers  
  
We try our best to make sure Metabase works in as many browsers as...

REQ-227: ---  
title: Upgrading Metabase  
redirect\_from:  
 - /docs/latest/operations-guide/upgrading-metabase  
---  
  
# Upgrading Metabase  
  
This page covers how to upgrade to a new Metabase release.  
  
-...

REQ-228: ---  
title: Account settings  
redirect\_from:  
 - /docs/latest/users-guide/account-settings  
---  
  
# Account settings  
  
You can view your account settings by going to the top right of the screen and...

REQ-229: ---  
title: API keys  
---  
  
# API keys  
  
Metabase can create API keys to authenticate programmatic requests to the API. To set the permissions for an API key, you can assign the key to a...

REQ-230: ---  
title: JWT-based authentication  
description: How to set up JWT-based authentication in Metabase to connect with your identity provider and manage user access.  
redirect\_from:  
 -...

REQ-231: ---  
title: SAML-based authentication  
redirect\_from:  
 - /docs/latest/enterprise-guide/authenticating-with-saml  
---  
  
# SAML-based authentication  
  
{% include plans-blockquote.html feature="SAML...

REQ-232: ---  
title: Passwords  
redirect\_from:  
 - /docs/latest/operations-guide/changing-password-complexity  
---  
  
# Passwords  
  
Metabase can allow authentication via email and password.  
  
## Password...

REQ-233: ---  
title: Session expiration  
redirect\_from:  
 - /docs/latest/operations-guide/changing-session-expiration  
---  
  
# Session expiration  
  
By default, Metabase sessions are valid for two weeks after a...

REQ-234: ---  
title: Google Sign-In  
redirect\_from:  
 - /docs/latest/administration-guide/10-single-sign-on  
 - /docs/latest/people-and-groups/google-and-ldap  
---  
  
# Google Sign-In  
  
Enabling [Google...

REQ-235: ---  
title: LDAP  
---  
  
# LDAP  
  
Metabase supports authentication with Lightweight Directory Access Protocol (LDAP).  
  
You can find SSO options under \*\*Admin settings\*\* > \*\*Settings\*\* >...

REQ-236: ---  
title: People and groups  
redirect\_from:  
 - /docs/latest/administration-guide/04-managing-users  
---  
  
# People and groups  
  
People can have [accounts](#creating-an-account) in Metabase, and those...

REQ-237: ---  
title: SAML with Auth0  
redirect\_from:  
 - /docs/latest/enterprise-guide/saml-auth0  
---  
  
# SAML with Auth0  
  
{% include plans-blockquote.html feature="SAML authentication" %}  
  
1. [Configure SAML in...

REQ-238: ---  
title: SAML with Microsoft Entra ID  
  
redirect\_from:  
 - /docs/latest/enterprise-guide/authenticating-with-saml-azure-ad  
---  
  
# SAML with Microsoft Entra ID  
  
{% include plans-blockquote.html...

REQ-239: ---  
title: SAML with Google  
redirect\_from:  
 - /docs/latest/enterprise-guide/saml-google  
---  
  
# SAML with Google  
  
{% include plans-blockquote.html feature="Google SAML authentication" %}  
  
1. Set up a...

REQ-240: ---  
title: SAML with Keycloak  
redirect\_from:  
 - /docs/latest/enterprise-guide/saml-keycloak  
---  
  
# SAML with Keycloak  
  
Keycloak is an open source platform that can be used as a user directory to...

REQ-241: ---  
title: SAML with Okta  
---  
  
# SAML with Okta  
  
{% include plans-blockquote.html feature="Okta SAML authentication" %}  
  
1. [Turn on SAML-based SSO in...

REQ-242: ---  
title: "People overview"  
redirect\_from:  
 - /docs/latest/administration-guide/sso  
 - /docs/latest/people-and-groups  
---  
  
# People overview  
  
User accounts, groups, and authentication. For...

REQ-243: ---  
title: User provisioning with SCIM  
---  
  
# User provisioning with SCIM  
  
{% include plans-blockquote.html feature="User provisioning with SCIM" %}  
  
Metabase supports user provisioning via the...

REQ-244: ---  
title: Application permissions  
redirect\_from:  
 - /docs/latest/administration-guide/application-permissions  
---  
  
# Application permissions  
  
{% include plans-blockquote.html feature="Application...

REQ-245: ---  
title: Collection permissions  
redirect\_from:  
 - /docs/latest/administration-guide/06-collections  
---  
  
# Collection permissions  
  
![Collection detail](./images/collection-detail.png)  
  
You can use...

REQ-246: ---  
title: Data permissions  
redirect\_from:  
 - /docs/latest/administration-guide/data-permissions  
---  
  
# Data permissions  
  
This page covers permissions for databases and tables. If you haven't...

REQ-247: ---  
title: Database routing  
summary: Route queries to different databases based on who's viewing them. Great for multi-tenant setups where each customer has their own database.  
---  
  
# Database...

REQ-248: ---  
title: Configuring permissions for embedding  
summary: Learn about which permissions tooling you should use depending on whether your customer data is in one database or split across multiple...

REQ-249: ---  
title: Impersonation access  
---  
  
# Impersonation permissions  
  
{% include plans-blockquote.html feature="Impersonation access" %}  
  
> For now, impersonation access is only available for ClickHouse,...

REQ-250: ---  
title: Permissions introduction  
redirect\_from:  
 - /docs/latest/administration-guide/05-setting-permissions  
---  
  
# Permissions introduction  
  
There are always going to be sensitive bits of...

REQ-251: ---  
title: Migrating from legacy permissions  
---  
  
# Migrating from legacy permissions  
  
> In Metabase 56, we renamed Data sandboxing to Row and column security. The functionality is the same. Some...

REQ-252: ---  
title: Notification permissions  
summary: Learn who can create and edit alerts and dashboard subscriptions, and what data recipients can see in their notifications.  
---  
  
# Notification...

REQ-253: ---  
title: Row and column security examples  
redirect\_from:  
 - /docs/latest/permissions/row-and-column-security-examples  
---  
  
# Row and column security examples  
  
{% include plans-blockquote.html...

REQ-254: ---  
title: Row and column security  
redirect\_from:  
 - /docs/latest/enterprise-guide/data-sandboxes  
 - /docs/latest/permissions/data-sandboxes  
---  
  
# Row and column security  
  
{% include...

REQ-255: ---  
title: Snippet folder permissions  
redirect\_from:  
 - /docs/latest/enterprise-guide/sql-snippets  
---  
  
# Snippet folder permissions  
  
{% include plans-blockquote.html feature="Snippet controls"...

REQ-256: ---  
title: Permissions overview  
redirect\_from:  
 - /docs/latest/permissions  
---  
  
# Permissions overview  
  
Metabase has a simple, but powerful permissions system.  
  
## [Permissions...

REQ-257: ---  
title: Alerts  
summary: Set up alerts to get notified via email, Slack, or webhooks when your questions return interesting results.  
redirect\_from:  
 - /docs/latest/users-guide/15-alerts  
 -...

REQ-258: ---  
title: Exporting results  
summary: Exporting your Metabase questions and dashboards as CSV, Excel, JSON, PDF, or PNG files.  
redirect\_from:  
 -...

REQ-259: ---  
title: Questions  
redirect\_from:  
 - /docs/latest/users-guide/04-asking-questions  
 - /docs/latest/users-guide/06-sharing-answers  
---  
  
# Questions  
  
![Metabase...

REQ-260: ---  
title: Basic SQL parameters  
summary: Text, number, and date variables let you plug basic values into your SQL code.  
---  
  
# Basic SQL parameters  
  
> If you want to filter on a database field in...

REQ-261: ---  
title: Field filters  
summary: Field filters let you create smart filter widgets for your SQL questions by connecting variables to database fields. Where possible, prefer field filters to basic...

REQ-262: ---  
title: Filter and parameter widgets for native code  
---  
  
# Filter and parameter widgets for native code  
  
When you add a [SQL variable or parameter](./sql-parameters.md) to your native/SQL query,...

REQ-263: ---  
title: Optional variables  
summary: Make parts of your SQL query optional by wrapping clauses in double brackets.  
---  
  
# Optional variables  
  
You can make a clause optional in a query. For example,...

REQ-264: ---  
title: Referencing models and saved questions  
redirect\_from:  
 - /docs/latest/users-guide/referencing-saved-questions-in-queries  
---  
  
# Referencing models and saved questions  
  
With SQL databases,...

REQ-265: ---  
title: Snippets  
redirect\_from:  
 - /docs/latest/users-guide/sql-snippets  
 - /docs/latest/questions/native-editor/sql-snippets  
---  
  
# Snippets  
  
![SQL...

REQ-266: ---  
title: SQL parameters  
summary: Create SQL templates by adding filters and parameters to your SQL questions in the native code editor.  
redirect\_from:  
 -...

REQ-267: ---  
title: Time grouping parameter  
---  
  
# Time grouping parameter  
  
You can add a parameter to SQL questions to change how results are grouped by time: by day, week, month, and so on.  
  
To add a time...

REQ-268: ---  
title: SQL editor  
redirect-from:  
 - /docs/latest/users-guide/writing-sql  
 - /docs/latest/questions/native-editor  
---  
  
# SQL editor  
  
If you ever need to ask questions that can't be expressed...

REQ-269: ---  
title: The query builder  
redirect\_from:  
 - /docs/latest/questions/query-builder/introduction  
 - /docs/latest/questions/query-builder  
---  
  
# The query builder  
  
Metabase includes a graphical...

REQ-270: ---  
title: Case  
---  
  
# Case  
  
`case` (alias `if`) checks if a value matches a list of conditions, and returns some output based on the first condition that's met. `if` and `case` work exactly the...

REQ-271: ---  
title: Coalesce  
---  
  
# Coalesce  
  
`coalesce` looks at the values in a list (in order), and returns the first non-null value.  
  
This function is useful when you want to:  
  
- [fill in missing...

REQ-272: ---  
title: Concat  
---  
  
# Concat  
  
`concat` concatenates data from two or more columns or values, and returns a string.  
  
## Syntax  
  
```  
concat(value1, value2, ...)  
```  
  
`value1`, `value2` ... can be...

REQ-273: ---  
title: ConvertTimezone  
---  
  
# ConvertTimezone  
  
`convertTimezone` shifts a timestamp into a specified time zone by adding or subtracting the right interval from the timestamp.  
  
| Syntax ...

REQ-274: ---  
title: CountIf  
---  
  
# CountIf  
  
`CountIf` counts the total number of rows in a table that match a condition. `CountIf` counts every row, not just unique rows.  
  
Syntax:...

REQ-275: ---  
title: Cumulative count and sum  
redirect\_from:  
 - /docs/latest/questions/expressions/cumulativesum  
 - /docs/latest/questions/expressions/cumulativecount  
---  
  
# Cumulative count and...

REQ-276: ---  
title: DatetimeAdd  
---  
  
# DatetimeAdd  
  
`datetimeAdd` takes a datetime value and adds some unit of time to it. This function is useful when you're working with time series data that's marked by a...

REQ-277: ---  
title: DatetimeDiff  
---  
  
# DatetimeDiff  
  
`datetimeDiff` gets the amount of time between two datetime values, using the specified unit of time. Note that the difference is calculated in \_whole\_...

REQ-278: ---  
title: DatetimeSubtract  
---  
  
# DatetimeSubtract  
  
`datetimeSubtract` takes a datetime value and subtracts some unit of time from it. You might want to use this function when working with time...

REQ-279: ---  
title: In  
---  
  
# In  
  
`in` compares values and returns true if `value1` equals `value2` (OR `value3`, etc., if specified).  
  
## Syntax  
  
```  
in(value1, value2, ...)  
```  
  
`value1` is the column or...

REQ-280: ---  
title: Isempty  
---  
  
# Isempty  
  
`isEmpty` checks whether a value in a \*\*string column\*\* is an empty string (`""`) or null. Calling `isEmpty` on a non-string column would cause an error.  
  
##...

REQ-281: ---  
title: Isnull  
---  
  
# Isnull  
  
`isNull` checks if a value is a `null`, a special kind of placeholder that's used by a database when something is missing or unknown.  
  
## Syntax  
  
```  
isNull(text...

REQ-282: ---  
title: Now  
---  
  
# Now  
  
`now` returns the current datetime using your Metabase [report timezone](../../../configuring-metabase/localization.md#report-timezone).  
  
## Creating conditional logic...

REQ-283: ---  
title: Offset  
---  
  
# Offset  
  
> The `Offset` function is currently unavailable for MySQL/MariaDB, ClickHouse, MongoDB, and Druid.  
  
The `Offset` function returns the value of an expression in a...

REQ-284: ---  
title: RegexExtract  
---  
  
# RegexExtract  
  
> `regexExtract` is unavailable for MongoDB, SQLite, and SQL Server. For Druid, `regexExtract` is only available for the Druid-JDBC...

REQ-285: ---  
title: Substring  
---  
  
# Substring  
  
`substring` extracts part of some text. This function is useful for cleaning up text (or any value with a [string data...

REQ-286: ---  
title: SumIf  
---  
  
# SumIf  
  
`SumIf` adds up the values in a column based on a condition.  
  
Syntax: `SumIf(column, condition)`.  
  
Example: in the table below, `SumIf([Payment], [Plan] = "Basic")`...

REQ-287: ---  
title: Week of year  
description: In Metabase, you can group by week of year in the query builder, or extract the week of year from a date column using a custom expression. Metabase supports...

REQ-288: ---  
title: List of expressions  
redirect\_from:  
 - /docs/latest/users-guide/expressions-list  
---  
  
# List of expressions  
  
For an introduction to expressions, check out the [overview of custom...

REQ-289: ---  
title: Custom expressions  
redirect\_from:  
 - /docs/latest/users-guide/expressions  
---  
  
# Custom expressions  
  
![Custom expression editor](../images/custom-expression-editor.png)  
  
[Custom...

REQ-290: ---  
title: Filtering  
---  
  
# Filtering  
  
Filtering just means narrowing things down based on certain criteria. You're probably already familiar with filtering when looking for something online, like...

REQ-291: ---  
title: Joining data  
summary: Learn how to combine data from different tables using joins. We'll show you how to pick tables, match columns, and choose the right join type.  
redirect\_from:  
 -...

REQ-292: ---  
title: Summarizing and grouping  
summary: Learn how to summarize and group your data in Metabase's query builder to calculate metrics like counts, sums, and averages across different...

REQ-293: ---  
title: Questions overview  
redirect\_from:  
 - /docs/latest/questions  
---  
  
# Questions overview  
  
Questions are queries plus their visualization. You can ask questions using Metabase's graphical...

REQ-294: ---  
title: Combo charts  
redirect\_from:  
 - /docs/latest/questions/sharing/visualizations/combo-chart  
---  
  
# Combo charts  
  
Combo charts let you combine bars and lines (or areas) on the same...

REQ-295: ---  
title: Country codes  
---  
  
# Country codes  
  
This reference lists all country codes and their corresponding country names used in Metabase's default world map visualizations. The data comes from...

REQ-296: ---  
title: Detail  
redirect\_from:  
 - /docs/latest/questions/sharing/visualizations/detail  
---  
  
# Detail  
  
The \*\*Detail\*\* visualization shows a single result record (row) in an easy-to-read, two-column...

REQ-297: ---  
title: Funnel charts  
redirect\_from:  
 - /docs/latest/questions/sharing/visualizations/funnel  
description: Funnel charts visualize how a value is broken out by a series of steps, and the percent...

REQ-298: ---  
title: Gauge chart  
redirect\_from:  
 - /docs/latest/questions/sharing/visualizations/gauge  
---  
  
# Gauge chart  
  
Ah, \*\*gauges\*\*: you either love 'em or you hate 'em. ...Or you feel "meh" about them,...

REQ-299: ---  
title: Line charts, bar charts, and area charts  
redirect\_from:  
 - /docs/latest/questions/visualizations/line-bar-and-area-charts  
---  
  
# Line charts, bar charts, and area charts  
  
They're pretty...

REQ-300: ---  
title: Maps  
redirect\_from:  
 - /docs/latest/questions/sharing/visualizations/maps  
description: "Maps in Metabase allow you to visualize geographical data either using coordinates or by region....

REQ-301: ---  
title: Numbers  
redirect\_from:  
 - /docs/latest/questions/sharing/visualizations/numbers  
---  
  
# Numbers  
  
The \*\*Numbers\*\* option is for displaying a single number, nice and big. The options for...

REQ-302: ---  
title: Pie and sunburst charts  
redirect\_from:  
 - /docs/latest/questions/sharing/visualizations/pie-or-donut-chart  
---  
  
# Pie and sunburst charts  
  
A \*\*pie chart\*\* can be used when breaking out a...

REQ-303: ---  
title: Pivot tables  
redirect\_from:  
 - /docs/latest/questions/sharing/visualizations/pivot-table  
---  
  
# Pivot tables  
  
> Pivot tables are currently only supported for questions built in the [query...

REQ-304: ---  
title: Progress bars  
redirect\_from:  
 - /docs/latest/questions/sharing/visualizations/progress-bar  
---  
  
# Progress bars  
  
\*\*Progress bars\*\* are for comparing a single number to a goal value that...

REQ-305: ---  
title: Sankey charts  
redirect\_from:  
 - /docs/latest/questions/sharing/visualizations/sankey  
---  
  
# Sankey charts  
  
Sankey charts show how data flows through multi-dimensional steps. They're...

REQ-306: ---  
title: Scatterplots and bubble charts  
redirect\_from:  
 - /docs/latest/questions/sharing/visualizations/scatterplot-or-bubble-chart  
---  
  
# Scatterplots and bubble charts  
  
\*\*Scatterplots\*\* are...

REQ-307: ---  
title: Tables  
redirect\_from:  
 - /docs/latest/questions/sharing/visualizations/table  
---  
  
# Tables  
  
![A table in Metabase](../images/table.png)  
  
Tables are data's natural habitat.  
  
##...

REQ-308: ---  
title: Tooltips  
redirect\_from:  
 - /docs/latest/questions/sharing/visualizations/tooltips  
---  
  
# Tooltips  
  
When you hover over a data point on a chart in Metabase, you'll see a tooltip about that...

REQ-309: ---  
title: Trend  
redirect\_from:  
 - /docs/latest/questions/sharing/visualizations/trend  
---  
  
# Trend  
  
![Trend settings](../images/trend-settings.png)  
  
The \*\*Trend\*\* visualization is great for...

REQ-310: ---  
title: Visualization overview  
redirect\_from:  
 - /docs/latest/users-guide/05-visualizing-results  
 - /docs/latest/questions/sharing/visualizing-results  
 -...

REQ-311: ---  
title: Waterfall charts  
redirect\_from:  
 - /docs/latest/questions/sharing/visualizations/waterfall-chart  
---  
  
# Waterfall charts  
  
Waterfall charts are a kind of bar chart useful for visualizing...

REQ-312: ---  
title: Troubleshooting BigQuery and Google Drive connections in Metabase  
---  
  
# Troubleshooting BigQuery and Google Drive connections in Metabase  
  
[This...

REQ-313: ---  
title: Reporting a bug  
---  
  
# Reporting a bug  
  
If you come across something that looks like a bug, please start by searching our [GitHub issues][metabase-issues] to see if it has already been...

REQ-314: ---  
title: People can't log in to Metabase  
---  
  
# People can't log in to Metabase  
  
## Reset password  
  
To reset a password for your Metabase instance, see:  
  
- [Reset a user's...

REQ-315: ---  
title: I can't see my tables  
---  
  
# I can't see my tables  
  
You've connected Metabase to a database, but:  
  
- you don't see the tables in the [Table Metadata](../data-modeling/metadata-editing.md)...

REQ-316: ---  
title: Metabase isn't sending email  
---  
  
# Metabase isn't sending email  
  
You have told Metabase to send email notifications, but:  
  
- the notifications aren't arriving.  
  
Before any other...

REQ-317: ---  
title: Can't view or edit  
---  
  
# Can't view or edit  
  
1. Clear your browser cache.  
2. Check if a browser extension or plugin is interfering with Metabase:  
 - Disable all extensions and plugins,  
...

REQ-318: ---  
title: Creating a HAR file for troubleshooting  
---  
  
# Creating a HAR file for troubleshooting  
  
HAR (short for \_HTTP Archive\_) files record network requests generated by your browser. HAR files...

REQ-319: ---  
title: Troubleshooting data permissions  
---  
  
# Troubleshooting data permissions  
  
If a person has the wrong level of access to the data that's returned by a question or query, you'll need to...

REQ-320: ---  
title: Troubleshooting database connections  
redirect\_from:  
 - /docs/latest/troubleshooting-guide/datawarehouse  
---  
  
# Troubleshooting database connections  
  
If you can't connect to your database,...

REQ-321: ---  
title: Troubleshooting database performance  
---  
  
# Troubleshooting database performance  
  
This guide deals with databases or data warehouses that are [connected to...

REQ-322: ---  
title: Diagnostic information for troubleshooting  
---  
  
# Diagnostic information for troubleshooting  
  
To download diagnostic information, hit `Cmd + F1` on Macs, `Ctrl + F1` on PCs. Hit Cmd/Ctrl +...

REQ-323: ---  
title: Troubleshooting Metabase on Docker  
---  
  
# Troubleshooting Metabase on Docker  
  
Docker simplifies many aspects of running Metabase, but there are some pitfalls to keep in mind. If you have...

REQ-324: ---  
title: Troubleshooting error messages  
---  
  
# Troubleshooting error messages  
  
An error message can help you find the right troubleshooting guide. The exact wording depends on your database and...

REQ-325: ---  
title: Troubleshooting filters  
---  
  
# Troubleshooting filters  
  
It's always a good idea to start with a quick sanity check:  
  
1. Clear your browser cache.  
2. Refresh the page.  
3.

REQ-326: ---  
title: Troubleshooting guides  
---  
  
# Troubleshooting guides  
  
Problems, their causes, how to detect them, and how to fix them.  
  
## Getting diagnostic info  
  
- [Download diagnostic...

REQ-327: ---  
title: How to find a known bug or limitation  
---  
  
# How to find a known bug or limitation  
  
If you can't find or solve your problem using the [troubleshooting guides](./index.md), you may be...

REQ-328: ---  
title: Troubleshooting LDAP  
---  
  
# Troubleshooting LDAP  
  
Metabase can use LDAP for authentication. [This article](../people-and-groups/ldap.md) explains how to set it up, and the guide below will...

REQ-329: ---  
title: My linked filters don't work  
---  
  
# My linked filters don't work  
  
You have created a [linked filter][linked-filter-gloss] so that (for example) if a dashboard contains both a "State" and a...

REQ-330: ---  
title: Using or migrating from an H2 application database  
---  
  
# Using or migrating from an H2 application database  
  
You've installed Metabase, but:  
  
- You're trying to migrate the application...

REQ-331: ---  
title: Troubleshooting models  
---  
  
# Troubleshooting models  
  
What kind of problem are you having with your [model][model-docs]?  
  
## Can't create a model  
  
If you don't see [the model...

REQ-332: ---  
title: My dashboard is slow  
---  
  
# My dashboard is slow  
  
First, you'll want to make sure your browser is on friendly terms with Metabase:  
  
- Clear your browser cache and disable all extensions...

REQ-333: ---  
title: Troubleshooting notifications  
---  
  
# Troubleshooting notifications  
  
Metabase is failing to send notifications like alerts or dashboard subscriptions.  
  
\*\*Root cause:\*\*  
  
When long running...

REQ-334: ---  
title: Troubleshooting permissions  
---  
  
# Troubleshooting permissions  
  
If someone has the wrong level of access to a dashboard or a question, the problem may be coming from group settings,...

REQ-335: ---  
title: Can't save questions or dashboards, or getting a blank page  
---  
  
# Can't save questions or dashboards, or getting a blank page  
  
If attempting to save a question or dashboard sometimes...

REQ-336: ---  
title: How to request new features  
---  
  
# How to request new features  
  
1. Check out the [issues in the github repo][github-issues] to make sure someone hasn't already requested the feature.  
2. If...

REQ-337: ---  
title: Troubleshooting row and column security access  
redirect\_from:  
 - /docs/latest/troubleshooting-guide/sandboxing  
---  
  
# Troubleshooting row and column security  
  
[Row and column...

REQ-338: ---  
title: Troubleshooting memory and JVM issues  
---  
  
# Troubleshooting memory and JVM issues  
  
Metabase runs on the Java Virtual Machine (JVM), and depending on how it's configured, it may use the...

REQ-339: ---  
title: Troubleshooting SAML authentication setup  
---  
  
# Troubleshooting SAML authentication setup  
  
{% include plans-blockquote.html feature="SAML authentication" %}  
  
Some common problems when...

REQ-340: ---  
title: How to read the server logs  
---  
  
# How to read the server logs  
  
Here's an example log from running a query:  
  
```  
2021-07-07 15:53:18,560 DEBUG middleware.log :: POST /api/dataset 202...

REQ-341: ---  
title: Troubleshooting SQL questions  
---  
  
# Troubleshooting SQL questions  
  
## Incorrect results  
  
- [Aggregations (counts, sums, etc.) are...

REQ-342: ---  
title: Troubleshooting syncs, scans, and fingerprinting  
---  
  
# Troubleshooting syncs, scans, and fingerprinting  
  
First, check if your data is outdated because of browser caching:  
  
1. Clear your...

REQ-343: ---  
title: Troubleshooting connection timeouts  
---  
  
# Troubleshooting connection timeouts  
  
If your queries are hanging or timing out, the problem could be coming from your:  
  
- [Database...

REQ-344: ---  
title: The dates and times in my questions and charts are wrong  
---  
  
# The dates and times in my questions and charts are wrong  
  
You are doing calculations with dates and times, or displaying...

REQ-345: ---  
title: Troubleshooting question and dashboard visualizations  
---  
  
# Troubleshooting question and dashboard visualizations  
  
To start, check if your current browser settings are compatible with...

REQ-346: ---  
title: Auditing tools  
redirect\_from:  
 - /docs/latest/enterprise-guide/audit  
---  
  
# Auditing tools  
  
> Auditing tools are deprecated. Instead, check out the (much better) [Usage...

REQ-347: ---  
title: "Tools overview"  
redirect\_from:  
 - /docs/latest/usage-and-performance-tools  
---  
  
# Tools overview  
  
Administration tools for managing your Metabase.  
  
## [Usage...

REQ-348: ---  
title: Admin tools  
redirect\_from:  
 - /docs/latest/enterprise-guide/tools  
---  
  
# Admin tools  
  
The Admin \*\*Tools\*\* tab contains features for troubleshooting. To get to the Admin tools sections, go...

REQ-349: ---  
title: Usage analytics  
---  
  
# Usage analytics  
  
{% include plans-blockquote.html feature="Usage analytics" %}  
  
The \*\*Usage analytics\*\* collection is a special collection that contains view-only...

REQ-350: Doc tools

REQ-351: import fs from 'fs';  
import path from 'path';  
import prettier from 'prettier';  
  
import { shortcuts } from '../../frontend/src/metabase/palette/shortcuts';  
  
interface Shortcut {  
 name: string;  
 ...

REQ-352: ---  
title: Keyboard shortcuts  
summary: A handy list of keyboard shortcuts to help you zip around Metabase.  
---  
  
# Keyboard shortcuts  
  
A handy list of keyboard shortcuts to help you zip around...

REQ-353: {  
 "rules": {  
 "no-unscoped-text-selectors": 2,  
 "import/no-commonjs": 0,  
 "no-color-literals": 0,  
 "no-console": 0,  
 "@typescript-eslint/no-namespace": "off",  
 ...

REQ-354: {  
 "rules": {  
 "import/no-default-export": "off",  
 "import/no-unresolved": "off"  
 }  
}

REQ-355: dependencies

REQ-356: Angular@20 Host App

REQ-357: import {  
 type ApplicationConfig,  
 provideZonelessChangeDetection,  
} from "@angular/core";  
import { provideRouter } from "@angular/router";  
import { defineMetabaseAuthConfig } from...

REQ-358: import type { Routes } from "@angular/router";  
  
import { InteractiveDashboardPageComponent } from "./interactive-dashboard-page.component";  
import { InteractiveQuestionPageComponent } from...

REQ-359: import {  
 type AfterViewInit,  
 Directive,  
 ElementRef,  
 Inject,  
 Input,  
 NgZone,  
 type OnChanges,  
 type OnDestroy,  
 type SimpleChanges,  
} from "@angular/core";  
import type { ReactElement }...

REQ-360: import { bootstrapApplication } from "@angular/platform-browser";  
  
import { AppComponent } from "./app/app.component";  
import { appConfig } from...

REQ-361: dependencies

REQ-362: dependencies

REQ-363: dependencies

REQ-364: import react from "@vitejs/plugin-react";  
import { defineConfig, loadEnv } from "vite";  
  
export default defineConfig(({ mode }) => {  
 const env = loadEnv(mode, process.cwd(), "");  
 const clientPort...

REQ-365: export const BACKEND\_PORT = process.env.BACKEND\_PORT ?? 4000;

REQ-366: export const SUCCESS\_EXIT\_CODE = 0;  
export const FAILURE\_EXIT\_CODE = 1;

REQ-367: import path from "path";  
  
export const ROOT\_FOLDER\_PATH = path.resolve(\_\_dirname, "../../..");  
export const E2E\_FOLDER\_PATH = path.join(ROOT\_FOLDER\_PATH, "e2e");

REQ-368: !/usr/bin/env node

REQ-369: const cypress = require("cypress");  
  
const { BACKEND\_PORT } = require("./constants/backend-port");  
const { FAILURE\_EXIT\_CODE } = require("./constants/exit-code");  
const { parseArguments, args } =...

REQ-370: const { execSync, spawn } = require("child\_process");  
  
const arg = require("arg");  
const chalk = require("chalk");  
const cypress = require("cypress");  
  
function printBold(message) {  
 ...

REQ-371: import path from "path";  
  
import { E2E\_FOLDER\_PATH } from "../../../constants/paths";  
  
export const HOST\_APP\_FOLDER\_PATH = path.join(  
 E2E\_FOLDER\_PATH,  
 "embedding-sdk-host-apps",  
);

REQ-372: import { BACKEND\_PORT } from "../../../constants/backend-port.js";  
  
const BASE\_ENV = {  
 WATCH: process.env.HOST\_APP\_ENVIRONMENT === "development" ? "true" : "false",  
 MB\_PORT: BACKEND\_PORT,  
 ...

REQ-373: import { shell } from "../../../cypress-runner-utils";  
import { waitForHealth } from "../../shared/helpers/wait-for-health";  
  
export async function startApp({  
 cwd,  
 env,  
 appRunCommand,  
 ...

REQ-374: import { startHostAppContainers } from "./start-host-app-containers";  
import type { HostAppTestSuiteName } from "./types";  
  
const testSuite = process.argv?.[2]?.trim() as HostAppTestSuiteName;  
  
if...

REQ-375: import { FAILURE\_EXIT\_CODE } from "../../constants/exit-code";  
import { printBold } from "../../cypress-runner-utils";  
import { setupAppCleanup } from "../shared/helpers/setup-app-cleanup";  
  
import {...

REQ-376: export type HostAppTestSuiteName =  
 | "vite-6-host-app-e2e"  
 | "next-15-app-router-host-app-e2e"  
 | "next-15-pages-router-host-app-e2e"  
 | "angular-20-host-app-e2e";

REQ-377: import path from "path";  
  
import { E2E\_FOLDER\_PATH } from "../../../constants/paths";  
  
export const E2E\_TMP\_FOLDER\_PATH = path.join(E2E\_FOLDER\_PATH, "tmp");

REQ-378: const BRANCH\_NAME = "main"; // Affects the `local` testing only. On CI is passed as an ENV variable.  
  
const BASE\_ENV = {  
 WATCH:  
 process.env.SAMPLE\_APP\_ENVIRONMENT === "development" ? "true" :...

REQ-379: import fs from "fs";  
import \* as path from "path";  
  
import { E2E\_TMP\_FOLDER\_PATH } from "../constants/e2e-tmp-folder-path";  
  
const LOCAL\_DIST\_PATH = "./local-dist";  
  
// See...

REQ-380: import fs from "fs";  
  
import { shell } from "../../../cypress-runner-utils";  
import { E2E\_TMP\_FOLDER\_PATH } from "../constants/e2e-tmp-folder-path";  
  
export function fetchApp({  
 appName,  
 ...

REQ-381: import fs from "fs";  
import \* as path from "path";  
  
import { ROOT\_FOLDER\_PATH } from "../../../constants/paths";  
  
const METABASE\_JAR\_DIST\_PATH = path.join(ROOT\_FOLDER\_PATH, "target/uberjar");  
  
const...

REQ-382: import { shell } from "../../../cypress-runner-utils";  
import { waitForHealth } from "../../shared/helpers/wait-for-health";  
  
export async function startContainers({  
 cwd,  
 env,  
 dockerUpCommand,  
...

REQ-383: import { startSampleAppContainers } from "./start-sample-app-containers";  
import type { SampleAppTestSuiteName } from "./types";  
  
const testSuite = process.argv?.[2]?.trim() as...

REQ-384: import { FAILURE\_EXIT\_CODE } from "../../constants/exit-code";  
import { printBold } from "../../cypress-runner-utils";  
import { setupAppCleanup } from "../shared/helpers/setup-app-cleanup";  
  
import {...

REQ-385: export type SampleAppTestSuiteName =  
 | "metabase-nodejs-react-sdk-embedding-sample-e2e"  
 | "metabase-nextjs-sdk-embedding-sample-e2e"  
 | "shoppy-e2e";  
  
export type EmbeddingSdkVersion = "local" |...

REQ-386: import fs from "fs";  
  
import { shell } from "../../../cypress-runner-utils";  
  
export function setupAppCleanup({  
 rootPath,  
 env,  
 appDownCommand,  
 cleanupAppDir,  
}: {  
 rootPath: string;  
 env:...

REQ-387: import { delay } from "../../../cypress-runner-utils";  
  
const HEALTH\_CHECK\_ATTEMPTS\_COUNT = 60 \* 5;  
const HEALTH\_CHECK\_WAIT\_TIME\_MS = 2000;  
  
export async function waitForHealth(url: string,...

REQ-388: const { FAILURE\_EXIT\_CODE } = require("./constants/exit-code");  
const CypressBackend = require("./cypress-runner-backend");  
const runCypress = require("./cypress-runner-run-tests");  
const { printBold...

REQ-389: import { FAILURE\_EXIT\_CODE, SUCCESS\_EXIT\_CODE } from "./constants/exit-code";  
import CypressBackend from "./cypress-runner-backend";  
import runCypress from "./cypress-runner-run-tests";  
import {  
 ...

REQ-390: import \_ from "underscore";  
  
import { loginCache } from "e2e/support/commands/user/authentication";  
import {  
 METABASE\_SECRET\_KEY,  
 SAMPLE\_DB\_ID,  
 SAMPLE\_DB\_TABLES,  
 USERS,  
 USER\_GROUPS,  
} from...

REQ-391: import {  
 addMongoDatabase,  
 addMySQLDatabase,  
 addPostgresDatabase,  
 restore,  
 setupWritableDB,  
 snapshot,  
} from "e2e/support/helpers";  
  
describe("qa databases snapshots", { tags: "@external"...

REQ-392: import fetch from "node-fetch"; // must be node-fetch v2 because it's non-esm  
  
const {  
 GITHUB\_RUN\_ID,  
 GITHUB\_TOKEN,  
 HASH,  
 PR\_NUMBER,  
 GITHUB\_REPOSITORY,  
 JOB\_NAME,  
} = process.env;  
  
const...

REQ-393: const fs = require("fs");  
const path = require("path");  
  
/\*\*  
 \* Collects failed test specs from the most recent Cypress test run  
 \*  
 \* After each run, a file will store failed spec paths as a...

REQ-394: We need to use the `mount` function from `@cypress/react` to allow  
running SDK component tests on multiple React versions

REQ-395: Cypress.Commands.add(  
 "addSQLiteDatabase",  
 ({ name = "sqlite", auto\_run\_queries = true, is\_full\_sync = true } = {}) => {  
 cy.log(`Add SQLite database DB called "${name}"`);  
 ...

REQ-396: const fs = require("fs");  
const path = require("path");  
  
const removeDirectory = (path) => {  
 try {  
 if (fs.existsSync(path)) {  
 fs.rmdirSync(path, { maxRetries: 10, recursive: true });  
 ...

REQ-397: export \* from "./downloadUtils-untyped";  
  
/\*\*  
 \* NOTE: Cypress commands are deprecated in favor of custom functions, as they  
 \* are easier to type. These commands rely on cypress tasks and were not...

REQ-398: Cypress.Commands.overwrite("log", (originalFn, text) => {  
 const logConfig = {  
 displayName: `${window.logCalls}. ${text}`,  
 name: "log",  
 message: "",  
 };  
  
 ...

REQ-399: import { SAMPLE\_DB\_TABLES, USER\_GROUPS } from "e2e/support/cypress\_data";  
  
const { STATIC\_ORDERS\_ID } = SAMPLE\_DB\_TABLES;  
  
const { COLLECTION\_GROUP } = USER\_GROUPS;  
  
Cypress.Commands.add(  
 ...

REQ-400: import type {  
 CollectionPermissions,  
 CollectionPermissionsGraph,  
 GroupsPermissions,  
 Impersonation,  
 PermissionsGraph,  
} from "metabase-types/api";  
  
declare global {  
 namespace Cypress {  
 ...

REQ-401: declare global {  
 namespace Cypress {  
 interface Chainable {  
 /\*\*  
 \* Get a button either unscoped, or chained to a previously yielded subject.  
 \* Uses `findByRole` under the...

REQ-402: import type { IconName } from "metabase/ui";  
  
declare global {  
 namespace Cypress {  
 interface Chainable {  
 /\*\*  
 \* Get an icon either unscoped, or chained to a previously yielded...

REQ-403: Cypress.Commands.add(  
 "paste",  
 { prevSubject: "element" },  
 (subject, text: string) => {  
 cy.wrap(subject).then(($element) => {  
 const clipboardData = new DataTransfer();  
 ...

REQ-404: import { USERS } from "e2e/support/cypress\_data";  
declare global {  
 namespace Cypress {  
 interface Chainable {  
 signIn: (user?: keyof typeof USERS, options?: LoginOptions) => void;  
 ...

REQ-405: import { USERS } from "e2e/support/cypress\_data";  
  
Cypress.Commands.add("createUserFromRawData", (user) => {  
 return cy.request("POST", "/api/user", user).then(({ body: user }) => {  
 // Dismiss...

REQ-406: declare global {  
 namespace Cypress {  
 interface Chainable {  
 findByTextEnsureVisible(  
 text: string,  
 ): Cypress.Chainable<JQuery<HTMLElement>>;  
 }  
 ...

REQ-407: Cypress.Commands.add(  
 "isRenderedWithinViewport",  
 {  
 prevSubject: true,  
 },  
 (subject) => {  
 const viewportTop = 0;  
 const viewportBottom = Cypress.$(cy.state("window")).height();  
 ...

REQ-408: import { POPOVER\_ELEMENT } from "e2e/support/helpers";  
  
Cypress.Commands.add(  
 "isVisibleInPopover",  
 {  
 prevSubject: true,  
 },  
 (subject) => {  
 cy.wrap(subject)  
 ...

REQ-409: this is the only place we allow direct helper import  
eslint-disable-next-line no-direct-helper-import

REQ-410: import "./cypress";  
import { renameConflictingCljsGlobals } from "metabase/embedding-sdk/test/rename-conflicting-cljs-globals";  
  
beforeEach(() => {  
 renameConflictingCljsGlobals();  
});

REQ-411: const fs = require("fs");  
const path = require("path");  
  
const webpack = require("webpack");  
  
const mainConfig = require("../../rspack.main.config");  
  
const SDK\_PACKAGE\_NAME =...

REQ-412: import fs from "node:fs";  
import path from "node:path";  
  
import installLogsPrinter from "cypress-terminal-report/src/installLogsPrinter";  
  
import \* as ciTasks from "./ci\_tasks";  
import {...

REQ-413: const { embeddingSdkComponentTestConfig } = require("./config");  
  
module.exports = {  
 component: embeddingSdkComponentTestConfig,  
};

REQ-414: const { defineConfig } = require("cypress");  
  
const { snapshotsConfig } = require("./config");  
  
module.exports = defineConfig({ e2e: snapshotsConfig });

REQ-415: const { defineConfig } = require("cypress");  
  
const { stressTestConfig } = require("./config");  
  
module.exports = defineConfig({ e2e: stressTestConfig });

REQ-416: const { defineConfig } = require("cypress");  
  
const { mainConfig } = require("./config");  
  
module.exports = defineConfig({ e2e: mainConfig });

REQ-417: import registerCypressGrep from "@cypress/grep"; // eslint-disable-line import/order  
registerCypressGrep();  
  
import "@cypress/skip-test/support";  
import...

REQ-418: import { ORDERS\_PRODUCTS\_ACCESS } from "./test\_roles";  
  
/\*\*  
 \* We are keeping the references to most commonly used ids and objects in this file.  
 \*  
 \* Please note that these ids are hard coded and...

REQ-419: \*

REQ-420: \*

REQ-421: import Knex from "knex";  
  
import { QA\_DB\_CONFIG, WRITABLE\_DB\_CONFIG } from "./cypress\_data";  
import { Roles } from "./test\_roles";  
import \* as testTables from "./test\_tables";  
  
const dbClients =...

REQ-422: !/usr/bin/env node

REQ-423: import type {  
 CardId,  
 Dashboard,  
 DashboardCard,  
 DashboardId,  
} from "metabase-types/api";  
  
import { DEFAULT\_CARD } from "./updateDashboardCards";  
  
export function addOrUpdateDashboardCard({  
 ...

REQ-424: import type { CardId, DashboardCard, DashboardId } from "metabase-types/api";  
  
export const addQuestionToDashboard = ({  
 dashboardId,  
 cardId,  
}: {  
 dashboardId: DashboardId;  
 cardId: CardId;  
}):...

REQ-425: import type { CollectionId } from "metabase-types/api";  
  
export const archiveCollection = (id: CollectionId) => {  
 cy.log(`Archiving a collection with id: ${id}`);  
  
 return cy.request("PUT",...

REQ-426: import type { Dashboard, DashboardId } from "metabase-types/api";  
  
export const archiveDashboard = (  
 id: DashboardId,  
): Cypress.Chainable<Cypress.Response<Dashboard>> => {  
 cy.log(`Archiving a...

REQ-427: import type { Card } from "metabase-types/api";  
  
export const archiveQuestion = (  
 id: Card["id"],  
): Cypress.Chainable<Cypress.Response<Card>> => {  
 cy.log(`Archiving a question with id:...

REQ-428: export const createApiKey = (name: string, group\_id: number) => {  
 return cy.request("POST", "/api/api-key", {  
 name,  
 group\_id,  
 });  
};

REQ-429: import type { Collection, RegularCollectionId } from "metabase-types/api";  
  
export const createCollection = ({  
 name,  
 description = null,  
 parent\_id = null,  
 authority\_level = null,  
 alias,  
}:...

REQ-430: import type {  
 CreateDashboardRequest,  
 Dashboard,  
 DashboardCard,  
} from "metabase-types/api";  
  
export interface DashboardDetails extends Omit<CreateDashboardRequest, "name"> {  
 name?: string;  
 ...

REQ-431: import type { Card, Dashboard, DashboardCard } from "metabase-types/api";  
  
import { cypressWaitAll } from "../e2e-misc-helpers";  
  
import { type DashboardDetails, createDashboard } from...

REQ-432: import type { Dashboard } from "metabase-types/api";  
  
import { type DashboardDetails, createDashboard } from "./createDashboard";  
  
export function createDashboardWithTabs({  
 dashcards = [],  
 tabs,  
...

REQ-433: import type { ModerationReview } from "metabase-types/api";  
  
export const createModerationReview = ({  
 status,  
 moderated\_item\_type,  
 moderated\_item\_id,  
}: {  
 status: "verified" | null;  
 ...

REQ-434: import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import type { Card, DatasetQuery, NativeQuery } from "metabase-types/api";  
  
import {  
 type Options,  
 type QuestionDetails,  
 logAction,  
 ...

REQ-435: import { createNativeQuestion } from "e2e/support/helpers";  
import type {  
 CardId,  
 Dashboard,  
 DashboardCard,  
 DashboardId,  
} from "metabase-types/api";  
  
import { type DashboardDetails,...

REQ-436: import type {  
 CardId,  
 CreateAlertNotificationRequest,  
 Notification,  
 NotificationCardSendCondition,  
 NotificationHandler,  
 UserId,  
} from "metabase-types/api";  
  
export const...

REQ-437: import type {  
 CreateSubscriptionRequest,  
 DashboardSubscription,  
} from "metabase-types/api";  
  
export const createPulse = ({  
 name = "Pulse",  
 cards = [],  
 channels = [],  
 dashboard\_id,  
}:...

REQ-438: import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import type {  
 Card,  
 DatasetQuery,  
 NativeQuery,  
 StructuredQuery,  
} from "metabase-types/api";  
  
import { visitMetric, visitModel,...

REQ-439: import type { Card, DashboardCard, DashboardId } from "metabase-types/api";  
  
import {  
 type NativeQuestionDetails,  
 createNativeQuestion,  
} from "./createNativeQuestion";  
import {  
 type...

REQ-440: import type { CardId, Dashboard, DashboardCard } from "metabase-types/api";  
  
import { type DashboardDetails, createDashboard } from "./createDashboard";  
import {  
 type StructuredQuestionDetails,  
 ...

REQ-441: export function createSnippet({  
 name,  
 content,  
}: {  
 name: string;  
 content: string;  
}): Cypress.Chainable<Cypress.Response<NativeQuerySnippet>> {  
 cy.request("POST",...

REQ-442: import type { CreateTimelineRequest, Timeline } from "metabase-types/api";  
  
export const createTimeline = ({  
 name = "Timeline",  
 icon = "star",  
 default: isDefault = false,  
 archived = false,  
 ...

REQ-443: import type {  
 CreateTimelineEventRequest,  
 TimelineEvent,  
} from "metabase-types/api";  
  
export const createTimelineEvent = ({  
 name = "Event",  
 icon = "star",  
 timestamp =...

REQ-444: import type {  
 CreateTimelineEventRequest,  
 CreateTimelineRequest,  
 Timeline,  
 TimelineEvent,  
} from "metabase-types/api";  
  
import { cypressWaitAll } from "../e2e-misc-helpers";  
  
import {...

REQ-445: import \_ from "underscore";  
  
import type { Dashboard, DashboardCard } from "metabase-types/api";  
  
export const editDashboardCard = (  
 dashboardCard: DashboardCard,  
 updatedProperties:...

REQ-446: import type { User } from "metabase-types/api";  
  
export const getCurrentUser = (): Cypress.Chainable<Cypress.Response<User>> => {  
 return cy.request<User>("GET", "/api/user/current");  
};

REQ-447: export { addOrUpdateDashboardCard } from "./addOrUpdateDashboardCard";  
export { addQuestionToDashboard } from "./addQuestionToDashboard";  
export { archiveCollection } from...

REQ-448: \*

REQ-449: import type { Dashboard, DashboardCard, DashboardId } from "metabase-types/api";  
  
export const DEFAULT\_CARD = {  
 id: -1,  
 row: 0,  
 col: 0,  
 size\_x: 11,  
 size\_y: 8,  
 visualization\_settings: {},  
...

REQ-450: import type { Settings } from "metabase-types/api";  
  
export const updateSetting = <  
 TKey extends keyof Settings,  
 TValue extends Settings[TKey],  
>(  
 setting: TKey,  
 value: TValue,  
):...

REQ-451: import { capitalize } from "inflection";  
  
import {  
 commandPalette,  
 commandPaletteButton,  
 commandPaletteInput,  
} from "./e2e-command-palette-helpers";  
import { NativeEditor } from...

REQ-452: import { SAMPLE\_DB\_ID, SAMPLE\_DB\_TABLES } from "e2e/support/cypress\_data";  
  
import { runNativeQuery } from "./e2e-misc-helpers";  
import { NativeEditor } from "./e2e-native-editor-helpers";  
  
const {  
 ...

REQ-453: eslint-disable-next-line no-direct-helper-import

REQ-454: import { queryBuilderMain, tableHeaderColumn } from "e2e/support/helpers";  
  
/\*\*  
 \* Initiate Summarize action  
 \*  
 \* @param {Object} options  
 \* @param {("notebook"|undefined)} options.mode  
 \*/  
export...

REQ-455: import { createNativeQuestion } from "./api";  
  
// until we have a test dataset that includes boolean data, we can use this questions to test booleans  
export function setupBooleanQuery(questionName =...

REQ-456: \*

REQ-457: import { updateSetting } from "./api";  
  
export const setupMetabaseCloud = () => {  
 updateSetting("site-url", "https://CYPRESSTESTENVIRONMENT.metabaseapp.com");  
};

REQ-458: import {  
 entityPickerModal,  
 entityPickerModalLevel,  
 entityPickerModalTab,  
 getFullName,  
 navigationSidebar,  
 popover,  
} from "e2e/support/helpers";  
import type { CollectionId } from...

REQ-459: export const commandPalette = () => cy.findByTestId("command-palette");  
export const shortcutModal = () =>  
 cy.findByRole("dialog", { name: "Shortcuts" });  
  
export const openCommandPalette = () =>...

REQ-460: import { popover } from "e2e/support/helpers/e2e-ui-elements-helpers";  
  
export function expressionEditorWidget() {  
 return cy.findByTestId("expression-editor");  
}  
  
export function...

REQ-461: import type {  
 DashCardId,  
 DashboardCard,  
 DashboardId,  
 DashboardTab,  
 VirtualDashboardCard,  
 WritebackActionId,  
} from "metabase-types/api";  
  
import { visitDashboard } from...

REQ-462: import type { VisualizationDisplay } from "metabase-types/api";  
  
import {  
 getDashboardCard,  
 showDashboardCardActions,  
} from "./e2e-dashboard-helpers";  
import { modal, sidebar } from...

REQ-463: import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import type {  
 Database,  
 DatabaseId,  
 FieldId,  
 TableId,  
} from "metabase-types/api";  
  
type FieldsMap = Record<string, FieldId>;  
type...

REQ-464: import type {  
 DatabaseId,  
 FieldId,  
 SchemaId,  
 TableId,  
} from "metabase-types/api";  
  
export const DataModel = {  
 visit,  
 get: getDataModel,  
 TablePicker: {  
 get: getTablePicker,  
 ...

REQ-465: export function getDimensions(isSelected) {  
 if (typeof isSelected === "undefined") {  
 return cy.findAllByTestId("dimension-list-item");  
 }  
  
 return cy.get(  
 ...

REQ-466: import xlsx, { type Sheet } from "xlsx";  
  
import {  
 getDashboardCard,  
 getDashboardCardMenu,  
 getEmbeddedDashboardCardMenu,  
} from "./e2e-dashboard-helpers";  
import { popover } from...

REQ-467: Rely on native drag events, rather than on the coordinates  
We have 3 "drag-handles" in this test. Their indexes are 0-based.

REQ-468: \*

REQ-469: import { openSharingMenu, sidebar } from "e2e/support/helpers";  
  
import { WEBMAIL\_CONFIG } from "../cypress\_data";  
  
const INBOX\_TIMEOUT = 5000;  
const INBOX\_INTERVAL = 100;  
  
const { WEB\_PORT,...

REQ-470: import { METABASE\_SECRET\_KEY } from "e2e/support/cypress\_data";  
import { modal, popover } from "e2e/support/helpers/e2e-ui-elements-helpers";  
  
import { openSharingMenu } from...

REQ-471: import { match } from "ts-pattern";  
  
import type { MetabaseTheme } from "metabase/embedding-sdk/theme/MetabaseTheme";  
import type { CreateApiKeyResponse } from "metabase-types/api";  
  
import {...

REQ-472: import { popover, tableInteractive } from "./e2e-ui-elements-helpers";  
  
export function assertSdkNotebookEditorUsable(root: Cypress.Chainable = cy) {  
 cy.findByText("Orders").should("be.visible");  
  
...

REQ-473: export const EMBEDDING\_SDK\_STORY\_HOST = "http://localhost:6006/iframe.html";  
  
export const getSdkRoot = () => cy.get("[data-cy-root]").should("be.visible");  
  
export const getStorybookSdkRoot = () =>  
...

REQ-474: export const mockSessionPropertiesTokenFeatures = (features) => {  
 cy.intercept({ method: "GET", url: "/api/session/properties" }, (request) => {  
 request.on("response", (response) => {  
 if...

REQ-475: import {  
 entityPickerModal,  
 modal,  
 popover,  
} from "e2e/support/helpers/e2e-ui-elements-helpers";  
  
export function setDropdownFilterType() {  
 cy.findByText("Dropdown list").click();  
}  
  
export...

REQ-476: import { AUTH\_PROVIDER\_URL } from "./embedding-sdk-helpers/constants";  
  
export const JWT\_SHARED\_SECRET = "0".repeat(64);  
  
export const enableJwtAuth = () => {  
 cy.request("PUT", "/api/setting", {  
 ...

REQ-477: import jwt from "jsonwebtoken";  
  
export function signJwt({  
 payload,  
 secret,  
}: {  
 payload: Record<string, string | number>;  
 secret: string;  
}): string {  
 return jwt.sign(payload, secret);  
}

REQ-478: \*

REQ-479: import type { StaticResponse } from "cypress/types/net-stubbing";  
  
import {  
 commandPaletteAction,  
 openCommandPalette,  
} from "./e2e-command-palette-helpers";  
import { appBar } from...

REQ-480: import { pickEntity } from "./e2e-collection-helpers";  
import { modal } from "./e2e-ui-elements-helpers";  
  
// Find a text field by label text, type it in, then blur the field.  
// Commonly used in our...

REQ-481: function mockProperty(propertyOrObject, value, url) {  
 cy.intercept("GET", url, (req) => {  
 req.reply((res) => {  
 if (typeof propertyOrObject === "object") {  
 Object.assign(res.body,...

REQ-482: export function createModelIndex({ modelId, pkName, valueName }) {  
 // since field ids are non-deterministic, we need to get them from the api  
 cy.request("GET",...

REQ-483: import {  
 interceptIfNotPreviouslyDefined,  
 popover,  
 tableInteractive,  
} from "e2e/support/helpers";  
  
export function datasetEditBar() {  
 return cy.findByTestId("dataset-edit-bar");  
}  
  
export...

REQ-484: import { popover } from "./e2e-ui-elements-helpers";  
  
function nativeEditor() {  
 cy.findAllByTestId("loading-indicator").should("not.exist");  
 return cy.get("[data-testid=native-query-editor]...

REQ-485: import type { CyHttpMessages } from "cypress/types/net-stubbing";  
  
import {  
 entityPickerModal,  
 entityPickerModalTab,  
 popover,  
 shouldDisplayTabs,  
} from...

REQ-486: import type { NotificationChannel } from "metabase-types/api/notification-channels";  
  
export const getAlertChannel = (name: string) =>  
 cy.findByRole("listitem", {  
 name,  
 });  
  
export const...

REQ-487: import \_ from "underscore";  
  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { modal, popover } from "e2e/support/helpers";  
  
export function selectSidebarItem(item) {  
 ...

REQ-488: import {  
 QA\_DB\_CONFIG,  
 QA\_DB\_CREDENTIALS,  
 QA\_MONGO\_PORT,  
 QA\_MYSQL\_PORT,  
 QA\_POSTGRES\_PORT,  
 WRITABLE\_DB\_CONFIG,  
 WRITABLE\_DB\_ID,  
} from "e2e/support/cypress\_data";  
  
import { createQuestion...

REQ-489: Units have different labels depending on the value and time direction (past/future)  
Example: "1 day", "2 days" for interval value "1 day ago" and "2 days from now" for offset value

REQ-490: \*

REQ-491: export function getSearchBar() {  
 return cy.findByPlaceholderText("Search...");  
}  
  
/\*\*  
 \* Checks the search results against expectedSearchValues, including descriptions,  
 \* collection names, and...

REQ-492: import { resetWritableDb } from "./e2e-qa-databases-helpers";  
  
export function snapshot(name) {  
 cy.request("POST", `/api/testing/snapshot/${name}`);  
}  
  
/\*\*  
 \*  
 \* @param { |  
 \* "blank" |  
 \* "setup"...

REQ-493: import { modal, popover } from "e2e/support/helpers/e2e-ui-elements-helpers";  
  
export const sharingMenuButton = () => cy.findByTestId("sharing-menu-button");  
export const sharingMenu = () =>...

REQ-494: Since there is no testing API token for Slack, it's easier to mock its configuration

REQ-495: import { updateSetting } from "e2e/support/helpers";  
  
const { IS\_ENTERPRISE } = Cypress.env();  
const HAS\_SNOWPLOW = Cypress.env("HAS\_SNOWPLOW\_MICRO");  
const SNOWPLOW\_URL =...

REQ-496: \*

REQ-497: import { match } from "ts-pattern";  
  
/\*\*  
 \* Just because an instance is using an Enterprise artifact (jar or Docker image),  
 \* doesn't mean that the token is active or that it has all feature flags...

REQ-498: Functions that get key elements in the app

REQ-499: export const isScrollableHorizontally = (element) => {  
 const { clientHeight, offsetHeight } = element;  
 const style = window.getComputedStyle(element);  
 const borderTopWidth =...

REQ-500: import { WRITABLE\_DB\_ID } from "../cypress\_data";  
  
import { modal } from "./e2e-ui-elements-helpers";  
  
export const FIXTURE\_PATH = "../../e2e/support/assets";  
  
export const VALID\_CSV\_FILES = [  
 {  
 ...

REQ-501: interface User {  
 email: string;  
 first\_name?: string | null;  
 last\_name?: string | null;  
}  
/\*\*  
 \* Get user's full name, or an email address if name is not available.  
 \*/  
export function...

REQ-502: import { popover } from "e2e/support/helpers/e2e-ui-elements-helpers";  
import { color as getColor } from "metabase/lib/colors";  
import { Icons } from "metabase/ui";  
import { GOAL\_LINE\_DASH } from...

REQ-503: export function openSeriesSettings(field, isBreakout = false) {  
 if (isBreakout) {  
 cy.get("[data-testid^=draggable-item]")  
 .contains(field)  
 .closest("[data-testid^=draggable-item]")  
...

REQ-504: export \* from "./component-embedding-sdk-helpers";  
export \* from "./component-embedding-sdk-question-helpers";  
export \* from "./component-embedding-sdk-console-helpers";

REQ-505: export const METABASE\_INSTANCE\_URL = "http://localhost:4000";  
export const AUTH\_PROVIDER\_URL = "http://auth-provider/sso";  
export const JWT\_SHARED\_SECRET =  
 ...

REQ-506: import \* as jose from "jose";  
  
import { loginCache } from "e2e/support/commands/user/authentication";  
import { USERS } from "e2e/support/cypress\_data";  
import {  
 AUTH\_PROVIDER\_URL,  
 ...

REQ-507: export \* from "./embedding-sdk-helpers";

REQ-508: export \* from "./api";  
export \* from "./e2e-action-helpers";  
export \* from "./e2e-ad-hoc-question-helpers";  
export \* from "./e2e-api-key-helpers";  
export \* from "./e2e-bi-basics-helpers";  
export \*...

REQ-509: H is for helpers

REQ-510: const { H } = cy;  
import { USERS } from "e2e/support/cypress\_data";  
  
import { setup } from "./visit-dashboard";  
  
describe("visitDashboard e2e helper", () => {  
 Object.keys(Cypress.\_.omit(USERS,...

REQ-511: import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 addOrUpdateDashboardCard,  
 createDashboard,  
 createNativeQuestion,  
 createNativeQuestionAndDashboard,  
 ...

REQ-512: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import type {  
 NativeQuestionDetails,  
 StructuredQuestionDetails,  
} from "e2e/support/helpers/api";  
import...

REQ-513: export const ORDERS\_PRODUCTS\_ACCESS = "orders\_products\_access";  
  
export const getCreatePostgresRoleIfNotExistSql = (roleName, grantSql) => {  
 return `  
 DO  
 $do$  
 BEGIN  
 IF NOT EXISTS (...

REQ-514: define test schema such that they can be used in multiple SQL dialects using  
knex's schema builder https://knexjs.org/guide/schema-builder.html  
we cannot use knex to define multi-dialect schemas...

REQ-515: export const many\_data\_types\_rows = [  
 {  
 uuid: "a0eebc99-9c0b-4ef8-bb6d-6bb9bd380a11",  
 integer: 1,  
 integerUnsigned: 2,  
 tinyint: -128,  
 tinyint1: 1,  
 smallint: 100,  
 ...

REQ-516: import dayjs from "dayjs";  
  
const { H } = cy;  
import { USER\_GROUPS, WRITABLE\_DB\_ID } from "e2e/support/cypress\_data";  
  
const WRITABLE\_TEST\_TABLE = "scoreboard\_actions";  
const FIRST\_SCORE\_ROW\_ID =...

REQ-517: import { assocIn } from "icepick";  
  
const { H } = cy;  
import { WRITABLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { many\_data\_types\_rows } from "e2e/support/test\_tables\_data";  
import {...

REQ-518: const { H } = cy;  
import { SAMPLE\_DB\_ID, WRITABLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 ORDERS\_DASHBOARD\_ID,  
 ...

REQ-519: import { assocIn } from "icepick";  
  
const { H } = cy;  
import {  
 SAMPLE\_DB\_ID,  
 USER\_GROUPS,  
 WRITABLE\_DB\_ID,  
} from "e2e/support/cypress\_data";  
import { IMPERSONATED\_USER\_ID } from...

REQ-520: const { H } = cy;  
import { SAMPLE\_DB\_ID, WRITABLE\_DB\_ID } from "e2e/support/cypress\_data";  
  
describe("issue 26470", { tags: "@external" }, () => {  
 beforeEach(() => {  
 H.restore("postgres\_12");  
 ...

REQ-521: import { QA\_MYSQL\_PORT, QA\_POSTGRES\_PORT } from "e2e/support/cypress\_data";  
  
import { waitForDbSync } from "./helpers/e2e-database-helpers";  
  
beforeEach(() => {  
 cy.H.restore();  
 ...

REQ-522: import \_ from "underscore";  
  
import {  
 QA\_POSTGRES\_PORT,  
 SAMPLE\_DB\_ID,  
 USER\_GROUPS,  
} from "e2e/support/cypress\_data";  
  
import { interceptPerformanceRoutes } from...

REQ-523: const { H } = cy;  
import \_ from "underscore";  
  
import { USER\_GROUPS } from "e2e/support/cypress\_data";  
import { DataPermissionValue } from "metabase/admin/permissions/types";  
  
import {...

REQ-524: const { H } = cy;  
import {  
 QA\_DB\_CREDENTIALS,  
 QA\_POSTGRES\_PORT,  
 USER\_GROUPS,  
} from "e2e/support/cypress\_data";  
import type { DatabaseData, User } from "metabase-types/api";  
  
const {...

REQ-525: import {  
 QA\_MONGO\_PORT,  
 QA\_MYSQL\_PORT,  
 QA\_POSTGRES\_PORT,  
 SAMPLE\_DB\_ID,  
 WRITABLE\_DB\_CONFIG,  
 WRITABLE\_DB\_ID,  
} from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from...

REQ-526: import {  
 SAMPLE\_DB\_ID,  
 SAMPLE\_DB\_SCHEMA\_ID,  
 USER\_GROUPS,  
 WRITABLE\_DB\_ID,  
} from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {...

REQ-527: const { H } = cy;  
import {  
 SAMPLE\_DB\_ID,  
 SAMPLE\_DB\_SCHEMA\_ID,  
 WRITABLE\_DB\_ID,  
} from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const {...

REQ-528: import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { H } = cy;  
const { ORDERS, ORDERS\_ID } = SAMPLE\_DATABASE;  
  
describe("scenarios > admin > datamodel > segments", () => {  
...

REQ-529: \*

REQ-530: import type { NonEmpty } from "metabase/i18n/types";  
import type { DictionaryArray } from "metabase-types/api";  
  
export const germanFieldNames: NonEmpty<DictionaryArray> = [  
 { locale: "de", msgid:...

REQ-531: import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 NORMAL\_USER\_ID,  
 ORDERS\_DASHBOARD\_ID,  
} from "e2e/support/cypress\_sample\_instance\_data";  
import type {...

REQ-532: import { parse } from "csv-parse/browser/esm/sync";  
  
import { METABASE\_SECRET\_KEY } from "e2e/support/cypress\_data";  
import type { DictionaryArray, DictionaryResponse } from...

REQ-533: import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import { NORMAL\_USER\_ID } from "e2e/support/cypress\_sample\_instance\_data";  
  
import { germanFieldNames } from...

REQ-534: import path from "path";  
  
import {  
 germanFieldNames,  
 invalidLocaleXX,  
 multipleInvalidLocales,  
 nonAsciiFieldNames,  
 portugueseFieldNames,  
 stringTranslatedTwice,  
} from "./constants";  
import...

REQ-535: import dayjs from "dayjs";  
import timezone from "dayjs/plugin/timezone";  
  
const { H } = cy;  
dayjs.extend(timezone);  
  
import {  
 freezeServerTime,  
 resetServerTime,  
} from...

REQ-536: const { H } = cy;  
  
import {  
 TEST\_TABLE,  
 instanceDefault,  
 sampleAdaptiveStrategy,  
 sampleDashboard,  
 sampleDatabase,  
 sampleDurationStrategy,  
 sampleQuestion,  
} from...

REQ-537: import { WRITABLE\_DB\_ID } from "e2e/support/cypress\_data";  
import type { NativeQuestionDetails } from "e2e/support/helpers";  
import {  
 type AdaptiveStrategy,  
 CacheDurationUnit,  
 type...

REQ-538: import { WRITABLE\_DB\_ID } from "e2e/support/cypress\_data";  
import type { NativeQuestionDetails } from "e2e/support/helpers";  
import {  
 createNativeQuestion,  
 createNativeQuestionAndDashboard,  
 ...

REQ-539: import { match } from "ts-pattern";  
  
import { popover } from "e2e/support/helpers";  
import {  
 type ScheduleComponentType,  
 getScheduleComponentLabel,  
} from...

REQ-540: import type { CacheStrategy, CacheableModel } from "metabase-types/api";  
  
export type CacheTestParameters = {  
 description: string;  
 strategy: CacheStrategy;  
 /\*\* Item whose cache we are testing...

REQ-541: const { H } = cy;  
  
import { WRITABLE\_DB\_ID } from "e2e/support/cypress\_data";  
  
import {  
 advanceServerClockBy,  
 interceptPerformanceRoutes,  
 resetServerTime,  
} from...

REQ-542: import type { ScheduleComponentType } from "metabase/common/components/Schedule/strings";  
import { checkNotNull } from "metabase/lib/types";  
import type { CacheableModel } from...

REQ-543: const { H } = cy;  
import {  
 ORDERS\_COUNT\_QUESTION\_ID,  
 ORDERS\_QUESTION\_ID,  
} from "e2e/support/cypress\_sample\_instance\_data";  
  
import {  
 goToPerformancePage,  
 interceptPerformanceRoutes,  
} from...

REQ-544: import dayjs from "dayjs";  
  
import { createMockTask } from "metabase-types/api/mocks";  
  
const { H } = cy;  
  
describe("scenarios > admin > tools > help", { tags: "@OSS" }, () => {  
 beforeEach(() => {  
...

REQ-545: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 ADMINISTRATORS\_GROUP\_ID,  
 ...

REQ-546: const { H } = cy;  
  
import { setupSaml } from "./sso/shared/helpers.js";  
  
describe("scenarios > admin > settings > user provisioning", () => {  
 beforeEach(() => {  
 H.restore();  
 ...

REQ-547: const { H } = cy;  
import { ORDERS\_DASHBOARD\_ID } from "e2e/support/cypress\_sample\_instance\_data";  
  
const downloadsFolder = Cypress.config("downloadsFolder");  
  
describe("error reporting modal", () =>...

REQ-548: import \_ from "underscore";  
  
const { H } = cy;  
import { USERS, USER\_GROUPS } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 ...

REQ-549: const { H } = cy;  
import {  
 SAMPLE\_DB\_ID,  
 SAMPLE\_DB\_SCHEMA\_ID,  
 WEBMAIL\_CONFIG,  
} from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {...

REQ-550: const { H } = cy;  
  
const CLIENT\_ID\_SUFFIX = "apps.googleusercontent.com";  
  
describe("scenarios > admin > settings > SSO > Google", () => {  
 beforeEach(() => {  
 H.restore();  
 ...

REQ-551: const { H } = cy;  
import { enableJwtAuth } from "e2e/support/helpers/e2e-jwt-helpers";  
  
import {  
 checkGroupConsistencyAfterDeletingMappings,  
 crudGroupMappingsWidget,  
} from...

REQ-552: const { H } = cy;  
  
import {  
 checkGroupConsistencyAfterDeletingMappings,  
 crudGroupMappingsWidget,  
} from "./shared/group-mappings-widget";  
  
describe(  
 "scenarios > admin > settings > SSO >...

REQ-553: const { H } = cy;  
  
import {  
 checkGroupConsistencyAfterDeletingMappings,  
 crudGroupMappingsWidget,  
} from "./shared/group-mappings-widget";  
import { getSamlCertificate, setupSaml } from...

REQ-554: import { popover } from "e2e/support/helpers";  
  
export function crudGroupMappingsWidget(authenticationMethod) {  
 cy.visit("/admin/settings/authentication/" + authenticationMethod);  
 ...

REQ-555: export function getSuccessUi() {  
 return cy.findByTestId("admin-layout-content").findByText("Success");  
}  
  
export const getSamlCertificate = () => {  
 return...

REQ-556: const { H } = cy;  
import { ORDERS\_QUESTION\_ID } from "e2e/support/cypress\_sample\_instance\_data";  
  
function checkFavicon(url) {  
 cy.request("/api/setting/application-favicon-url")  
 .its("body")  
 ...

REQ-557: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS\_ID, ORDERS, PEOPLE\_ID, PEOPLE,...

REQ-558: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS, ORDERS\_ID } =...

REQ-559: const { H } = cy;  
  
import { LONGITUDE\_OPTIONS } from "./shared/constants";  
  
describe("scenarios > binning > correctness > longitude", () => {  
 beforeEach(() => {  
 H.restore();  
 ...

REQ-560: export const TIME\_OPTIONS = {  
 Minute: {  
 selected: "by minute",  
 representativeValues: ["April 30, 2022, 6:56 PM", "May 10, 2022, 9:38 AM"],  
 },  
 Hour: {  
 selected: "by hour",  
 ...

REQ-561: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
import { TIME\_OPTIONS } from "./shared/constants";  
  
const { ORDERS\_ID } = SAMPLE\_DATABASE;  
  
const...

REQ-562: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS\_ID, ORDERS, PEOPLE\_ID, PEOPLE, PRODUCTS\_ID, PRODUCTS } =  
 SAMPLE\_DATABASE;  
  
/\*\*  
 \* The list...

REQ-563: const { H } = cy;  
import { ORDERS\_QUESTION\_ID } from "e2e/support/cypress\_sample\_instance\_data";  
  
/\*\*  
 \* The list of issues this spec covers:  
 \* - metabase#15648  
 \* -  
 \*/  
  
describe("scenarios >...

REQ-564: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS\_ID, PEOPLE\_ID } = SAMPLE\_DATABASE;  
  
describe("scenarios > binning > binning options", () => {  
...

REQ-565: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS, ORDERS\_ID } = SAMPLE\_DATABASE;  
  
const CREATED\_AT\_BREAKOUT = [  
 "field",  
 ...

REQ-566: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { PRODUCTS\_ID, PRODUCTS } =...

REQ-567: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
  
const questionDetails = {  
 name: "SQL Binning",  
 native: {  
 query:  
 "SELECT ORDERS.CREATED\_AT, ORDERS.TOTAL,...

REQ-568: import dayjs from "dayjs";  
import { assocIn } from "icepick";  
import { P, isMatching } from "ts-pattern";  
  
const { H } = cy;  
import { SAMPLE\_DB\_TABLES } from "e2e/support/cypress\_data";  
import {  
 ...

REQ-569: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 ORDERS\_COUNT\_QUESTION\_ID,  
 ORDERS\_DASHBOARD\_ID,  
 ORDERS\_QUESTION\_ID,  
} from...

REQ-570: const { H } = cy;  
import {  
 ADMIN\_PERSONAL\_COLLECTION\_ID,  
 FIRST\_COLLECTION\_ID,  
 ORDERS\_COUNT\_QUESTION\_ID,  
 ORDERS\_QUESTION\_ID,  
} from "e2e/support/cypress\_sample\_instance\_data";  
  
describe("issue...

REQ-571: import { assocIn } from "icepick";  
import \_ from "underscore";  
  
const { H } = cy;  
import { SAMPLE\_DB\_ID, USERS, USER\_GROUPS } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from...

REQ-572: export function displaySidebarChildOf(collectionName) {  
 cy.findByText(collectionName)  
 .parentsUntil("[data-testid=sidebar-collection-link-root]")  
 .find(".Icon-chevronright")  
 .eq(0) //...

REQ-573: const { H } = cy;  
import {  
 ORDERS\_DASHBOARD\_ID,  
 ORDERS\_QUESTION\_ID,  
} from "e2e/support/cypress\_sample\_instance\_data";  
  
const ANALYTICS\_COLLECTION\_NAME = "Usage analytics";  
const...

REQ-574: import { onlyOn } from "@cypress/skip-test";  
  
const { H } = cy;  
import { USERS } from "e2e/support/cypress\_data";  
import { FIRST\_COLLECTION\_ID } from...

REQ-575: const { H } = cy;  
import { USERS } from "e2e/support/cypress\_data";  
import {  
 ADMIN\_PERSONAL\_COLLECTION\_ID,  
 NORMAL\_USER\_ID,  
 NO\_DATA\_PERSONAL\_COLLECTION\_ID,  
} from...

REQ-576: import { onlyOn } from "@cypress/skip-test";  
  
const { H } = cy;  
import {  
 ORDERS\_DASHBOARD\_ID,  
 ORDERS\_QUESTION\_ID,  
} from "e2e/support/cypress\_sample\_instance\_data";  
  
const PERMISSIONS = {  
 ...

REQ-577: import { P, isMatching } from "ts-pattern";  
  
const { H } = cy;  
import {  
 FIRST\_COLLECTION\_ID,  
 ORDERS\_COUNT\_QUESTION\_ID,  
 ORDERS\_QUESTION\_ID,  
 READ\_ONLY\_PERSONAL\_COLLECTION\_ID,  
} from...

REQ-578: const { H } = cy;  
import { USER\_GROUPS, WRITABLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { FIRST\_COLLECTION\_ID } from "e2e/support/cypress\_sample\_instance\_data";  
import { FIXTURE\_PATH,...

REQ-579: export function parseVersionString(versionString) {  
 if (typeof versionString === "undefined") {  
 return []; // Return empty array if versionString is undefined  
 }  
  
 const segments =...

REQ-580: import {  
 setupInstance,  
 setupLanguage,  
} from "e2e/test/scenarios/cross-version/helpers/cross-version-helpers.js";  
  
import { version } from...

REQ-581: const { H } = cy;  
  
it("should generate metadata", () => {  
 cy.signInAsAdmin();  
  
 H.withSampleDatabase((SAMPLE\_DATABASE) => {  
 cy.writeFile("e2e/support/cypress\_sample\_database.json",...

REQ-582: import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 createMetric,  
 createSegment,  
} from...

REQ-583: const { H } = cy;  
import {  
 fillAreaUnderLineChart,  
 newQuestion,  
 saveQuestion,  
} from "e2e/test/scenarios/cross-version/helpers/cross-version-helpers.js";  
  
import { version } from...

REQ-584: import { parseVersionString } from "e2e/test/scenarios/cross-version/helpers/cross-version-helpers.js";  
  
export const version = parseVersionString(Cypress.env("SOURCE\_VERSION"));

REQ-585: const { defineConfig } = require("cypress");  
  
const { crossVersionSourceConfig } = require("e2e/support/config");  
  
module.exports = defineConfig({ e2e: crossVersionSourceConfig });

REQ-586: import { parseVersionString } from "e2e/test/scenarios/cross-version/helpers/cross-version-helpers.js";  
  
export const version = parseVersionString(Cypress.env("TARGET\_VERSION"));

REQ-587: const { defineConfig } = require("cypress");  
  
const { crossVersionTargetConfig } = require("e2e/support/config");  
  
module.exports = defineConfig({ e2e: crossVersionTargetConfig });

REQ-588: import {  
 assertTimelineData,  
 dismissOkToPlayWithQuestionsModal,  
} from "e2e/test/scenarios/cross-version/helpers/cross-version-helpers";  
import { version } from...

REQ-589: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import type {  
 DashboardDetails,  
 StructuredQuestionDetails,  
} from "e2e/support/helpers";  
import type {...

REQ-590: import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
  
const { H } = cy;  
  
type CastTestCase = {  
 name: string;  
 expression: string;  
 filterOperator: string;  
 filterValue: string;  
 ...

REQ-591: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import type { StructuredQuestionDetails } from "e2e/support/helpers";  
  
const { ORDERS\_ID, ORDERS } =...

REQ-592: const { H } = cy;  
  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
const { PRODUCTS, PRODUCTS\_ID } =...

REQ-593: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
  
describe("scenarios > question > custom column > expression shortcuts > combine", () => {  
 beforeEach(() => {  
 ...

REQ-594: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS\_ID, ORDERS } =...

REQ-595: const { H } = cy;  
  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS, ORDERS\_ID } = SAMPLE\_DATABASE;  
  
describe("scenarios > question > custom column > typing...

REQ-596: const { H } = cy;  
import { dedent } from "ts-dedent";  
  
import { SAMPLE\_DB\_ID, WRITABLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from...

REQ-597: const { H } = cy;  
import { dedent } from "ts-dedent";  
  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS,...

REQ-598: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 ADMIN\_PERSONAL\_COLLECTION\_ID,  
 ...

REQ-599: import { onlyOn } from "@cypress/skip-test";  
  
const { H } = cy;  
import { USERS } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {...

REQ-600: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import \* as S from "e2e/support/cypress\_sample\_instance\_data";  
import { createMockDashboardCard } from...

REQ-601: import { assoc } from "icepick";  
import \_ from "underscore";  
  
const { H } = cy;  
import { SAMPLE\_DB\_ID, USERS, USER\_GROUPS } from "e2e/support/cypress\_data";  
import {  
 ORDERS\_COUNT\_QUESTION\_ID,  
 ...

REQ-602: import { assoc } from "icepick";  
import \_ from "underscore";  
  
const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from...

REQ-603: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 ADMIN\_PERSONAL\_COLLECTION\_ID,  
 NORMAL\_PERSONAL\_COLLECTION\_ID,  
 ORDERS\_BY\_YEAR\_QUESTION\_ID,  
 ...

REQ-604: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import { ORDERS\_DASHBOARD\_ID } from "e2e/support/cypress\_sample\_instance\_data";  
import { createMockParameter }...

REQ-605: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS, ORDERS\_ID, PEOPLE, PEOPLE\_ID, PRODUCTS, PRODUCTS\_ID } =  
 ...

REQ-606: const { H } = cy;  
  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { ORDERS\_DASHBOARD\_ID } from "e2e/support/cypress\_sample\_instance\_data";  
import {  
 ORDERS\_COUNT\_BY\_CREATED\_AT,  
 ...

REQ-607: const { H } = cy;  
  
import { ORDERS\_DASHBOARD\_ID } from "e2e/support/cypress\_sample\_instance\_data";  
import {  
 ORDERS\_COUNT\_BY\_CREATED\_AT,  
 ORDERS\_COUNT\_BY\_CREATED\_AT\_AND\_PRODUCT\_CATEGORY,  
 ...

REQ-608: const { H } = cy;  
  
import { ORDERS\_DASHBOARD\_ID } from "e2e/support/cypress\_sample\_instance\_data";  
import {  
 ACCOUNTS\_COUNT\_BY\_COUNTRY,  
 COUNTRY\_CODES,  
 ORDERS\_COUNT\_BY\_PRODUCT\_CATEGORY,  
} from...

REQ-609: const { H } = cy;  
  
import {  
 ORDERS\_COUNT\_BY\_CREATED\_AT,  
 ORDERS\_COUNT\_BY\_PRODUCT\_CATEGORY,  
 PRODUCTS\_COUNT\_BY\_CATEGORY,  
 PRODUCTS\_COUNT\_BY\_CATEGORY\_PIE,  
 PRODUCTS\_COUNT\_BY\_CREATED\_AT,  
 ...

REQ-610: const { H } = cy;  
  
import {  
 PRODUCTS\_AVERAGE\_BY\_CATEGORY,  
 PRODUCTS\_COUNT\_BY\_CATEGORY,  
} from "e2e/support/test-visualizer-data";  
  
describe("scenarios > dashboard > visualizer > filters", () => {  
...

REQ-611: const { H } = cy;  
  
import { ORDERS\_DASHBOARD\_ID } from "e2e/support/cypress\_sample\_instance\_data";  
import {  
 ORDERS\_COUNT\_BY\_CREATED\_AT,  
 ORDERS\_COUNT\_BY\_PRODUCT\_CATEGORY,  
 ...

REQ-612: const { H } = cy;  
  
import {  
 ORDERS\_COUNT\_BY\_CREATED\_AT,  
 ORDERS\_COUNT\_BY\_PRODUCT\_CATEGORY,  
 PRODUCTS\_COUNT\_BY\_CATEGORY,  
 PRODUCTS\_COUNT\_BY\_CATEGORY\_PIE,  
 PRODUCTS\_COUNT\_BY\_CREATED\_AT,  
 ...

REQ-613: import { ORDERS\_DASHBOARD\_ID } from "e2e/support/cypress\_sample\_instance\_data";  
import {  
 ACCOUNTS\_COUNT\_BY\_CREATED\_AT,  
 ORDERS\_COUNT\_BY\_CREATED\_AT,  
 ORDERS\_COUNT\_BY\_PRODUCT\_CATEGORY,  
 ...

REQ-614: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import { ORDERS\_BY\_YEAR\_QUESTION\_ID } from...

REQ-615: const { H } = cy;  
import { SAMPLE\_DB\_ID, USER\_GROUPS } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 NORMAL\_USER\_ID,  
 ...

REQ-616: const { H } = cy;  
import {  
 ORDERS\_BY\_YEAR\_QUESTION\_ID,  
 ORDERS\_COUNT\_QUESTION\_ID,  
} from "e2e/support/cypress\_sample\_instance\_data";  
  
const cards = [  
 {  
 card\_id: ORDERS\_COUNT\_QUESTION\_ID,  
 ...

REQ-617: const { H } = cy;  
import { SAMPLE\_DB\_ID, WRITABLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import { ORDERS\_QUESTION\_ID } from...

REQ-618: import \_ from "underscore";  
  
const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import { GRID\_WIDTH }...

REQ-619: const { H } = cy;  
  
describe("scenarios > dashboard cards > undo", () => {  
 beforeEach(() => {  
 H.restore();  
 cy.signInAsAdmin();  
 });  
  
 it(  
 "when undoing a dashcard removal or dashcard...

REQ-620: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 ORDERS\_DASHBOARD\_DASHCARD\_ID,  
 ...

REQ-621: import {  
 ORDERS\_DASHBOARD\_ID,  
 READ\_ONLY\_PERSONAL\_COLLECTION\_ID,  
} from "e2e/support/cypress\_sample\_instance\_data";  
import \* as H from "e2e/support/helpers";  
import { createMockParameter } from...

REQ-622: const { H } = cy;  
import { USER\_GROUPS } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 FIRST\_COLLECTION\_ID,  
 ...

REQ-623: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 createMockDashboardCard,  
 createMockParameter,  
} from "metabase-types/api/mocks";  
  
const {...

REQ-624: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import { ORDERS\_DASHBOARD\_ID } from "e2e/support/cypress\_sample\_instance\_data";  
  
describe("scenarios >...

REQ-625: const { H } = cy;  
import { WRITABLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import { ORDERS\_DASHBOARD\_ID } from...

REQ-626: const { H } = cy;  
import { ORDERS\_DASHBOARD\_ID } from "e2e/support/cypress\_sample\_instance\_data";  
  
function filterDashboard(suggests = true) {  
 H.visitDashboard(ORDERS\_DASHBOARD\_ID);  
 ...

REQ-627: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import type { StructuredQuestionDetails } from "e2e/support/helpers";  
  
const { PRODUCTS\_ID, PRODUCTS } =...

REQ-628: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { PRODUCTS, PRODUCTS\_ID } = SAMPLE\_DATABASE;  
  
const FILTER = {  
 name: "Category",  
 slug: "category",  
...

REQ-629: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 ORDERS\_BY\_YEAR\_QUESTION\_ID,  
 ORDERS\_COUNT\_QUESTION\_ID,  
} from...

REQ-630: const { H } = cy;  
  
import { WRITABLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import type {  
 DashboardDetails,  
 ...

REQ-631: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS\_ID } = SAMPLE\_DATABASE;  
  
describe("dashboard filters values source config clearing and...

REQ-632: const { H } = cy;  
import {  
 ORDERS\_DASHBOARD\_DASHCARD\_ID,  
 ORDERS\_DASHBOARD\_ID,  
} from "e2e/support/cypress\_sample\_instance\_data";  
  
import \* as DateFilter from...

REQ-633: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS, ORDERS\_ID, PRODUCTS, PRODUCTS\_ID } = SAMPLE\_DATABASE;  
  
const questionDetails = {  
 name:...

REQ-634: const { H } = cy;  
import { ORDERS\_DASHBOARD\_ID } from "e2e/support/cypress\_sample\_instance\_data";  
  
import { addWidgetStringFilter } from...

REQ-635: const { H } = cy;  
import {  
 ORDERS\_DASHBOARD\_DASHCARD\_ID,  
 ORDERS\_DASHBOARD\_ID,  
} from "e2e/support/cypress\_sample\_instance\_data";  
  
import {  
 addWidgetStringFilter,  
 selectFilterValueFromList,  
}...

REQ-636: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import { createMockParameter } from "metabase-types/api/mocks";  
  
const { PEOPLE, PEOPLE\_ID, ORDERS\_ID } =...

REQ-637: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { PRODUCTS\_ID } = SAMPLE\_DATABASE;  
  
describe("scenarios > dashboard > filters > nested questions", ()...

REQ-638: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import { ORDERS\_DASHBOARD\_ID } from "e2e/support/cypress\_sample\_instance\_data";  
  
const { ACCOUNTS, ORDERS\_ID }...

REQ-639: const { H } = cy;  
import { ORDERS\_DASHBOARD\_ID } from "e2e/support/cypress\_sample\_instance\_data";  
  
import { addWidgetNumberFilter } from "../native-filters/helpers/e2e-field-filter-helpers";  
  
import...

REQ-640: const { H } = cy;  
  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import type {  
 DashboardDetails,  
 NativeQuestionDetails,  
 StructuredQuestionDetails,  
} from...

REQ-641: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 ORDERS\_COUNT\_QUESTION\_ID,  
 ORDERS\_QUESTION\_ID,  
} from...

REQ-642: const { H } = cy;  
import { USER\_GROUPS, WRITABLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { PRODUCTS, PRODUCTS\_ID } =...

REQ-643: const { H } = cy;  
  
import \* as DateFilter from "../native-filters/helpers/e2e-date-filter-helpers";  
  
import {  
 DASHBOARD\_SQL\_DATE\_FILTERS,  
 questionDetails,  
} from...

REQ-644: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
import { addWidgetStringFilter } from "../native-filters/helpers/e2e-field-filter-helpers";  
  
const { ORDERS }...

REQ-645: const { H } = cy;  
  
import {  
 addWidgetStringFilter,  
 applyFilterByType,  
 selectFilterValueFromList,  
} from "../native-filters/helpers/e2e-field-filter-helpers";  
  
import {  
 ...

REQ-646: const { H } = cy;  
  
describe("scenarios > dashboard > filters > SQL > management", () => {  
 beforeEach(() => {  
 H.restore();  
 cy.signInAsAdmin();  
 });  
  
 describe("number filter", () => {  
 ...

REQ-647: const { H } = cy;  
  
import { addWidgetNumberFilter } from "../native-filters/helpers/e2e-field-filter-helpers";  
  
import {  
 DASHBOARD\_SQL\_NUMBER\_FILTERS,  
 questionDetails,  
} from...

REQ-648: import { produce } from "immer";  
  
const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { PRODUCTS } = SAMPLE\_DATABASE;  
  
const questionDetails = {  
 name:...

REQ-649: const { H } = cy;  
  
const questionDetails = {  
 name: "Return input value",  
 native: {  
 query: "select {{filter}}",  
 "template-tags": {  
 filter: {  
 id:...

REQ-650: const { H } = cy;  
  
import { applyFilterByType } from "../native-filters/helpers/e2e-field-filter-helpers";  
  
import {  
 DASHBOARD\_SQL\_TEXT\_FILTERS,  
 questionDetails,  
} from...

REQ-651: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
import {  
 applyFilterByType,  
 selectDefaultValueFromPopover,  
} from...

REQ-652: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { PEOPLE, PEOPLE\_ID, PRODUCTS, PRODUCTS\_ID } = SAMPLE\_DATABASE;  
  
// the dashboard parameters used in...

REQ-653: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 ORDERS\_BY\_YEAR\_QUESTION\_ID,  
 ...

REQ-654: export const DASHBOARD\_DATE\_FILTERS = {  
 "Month and Year": {  
 value: {  
 month: "Nov",  
 year: "2022",  
 },  
 representativeResult: "85.88",  
 },  
 "Quarter and Year": {  
 value: {  
...

REQ-655: export const DASHBOARD\_LOCATION\_FILTERS = {  
 Is: {  
 value: "Abbeville",  
 representativeResult: "1510",  
 },  
 "Is not": {  
 value: "Abbeville",  
 representativeResult: "37.65",  
 },  
 ...

REQ-656: export const DASHBOARD\_NUMBER\_FILTERS = [  
 {  
 operator: "Equal to",  
 value: "2.07",  
 representativeResult: "37.65",  
 },  
 {  
 operator: "Equal to",  
 value: "2.07",  
 ...

REQ-657: import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { PEOPLE } = SAMPLE\_DATABASE;  
  
export const DASHBOARD\_SQL\_DATE\_FILTERS = {  
 "Month and Year": {  
 sqlFilter:...

REQ-658: import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { PEOPLE } = SAMPLE\_DATABASE;  
  
export const questionDetails = {  
 name: "SQL with Field Filter",  
 native: {  
 query:  
 ...

REQ-659: import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { PRODUCTS } = SAMPLE\_DATABASE;  
  
export const DASHBOARD\_SQL\_NUMBER\_FILTERS = {  
 "Equal to": {  
 sqlFilter:...

REQ-660: import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { PRODUCTS } = SAMPLE\_DATABASE;  
  
export const DASHBOARD\_SQL\_TEXT\_FILTERS = {  
 Is: {  
 sqlFilter: "string/=",  
 ...

REQ-661: export const DASHBOARD\_TEXT\_FILTERS = [  
 {  
 operator: "Is",  
 single: true,  
 value: "Organic",  
 representativeResult: "39.58",  
 },  
 {  
 operator: "Is not",  
 single: true,  
 ...

REQ-662: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS, ORDERS\_ID, PRODUCTS } = SAMPLE\_DATABASE;  
  
const dashboardDetails = {  
 name: "Test...

REQ-663: const { H } = cy;  
  
import \* as QSHelpers from "./shared/dashboard-filters-query-stages";  
  
/\*\*  
 \* Abbreviations used for card aliases in this test suite:  
 \* qbq = question-based question  
 \* qbm =...

REQ-664: const { H } = cy;  
  
import \* as QSHelpers from "./shared/dashboard-filters-query-stages";  
  
/\*\*  
 \* Abbreviations used for card aliases in this test suite:  
 \* qbq = question-based question  
 \* qbm =...

REQ-665: const { H } = cy;  
  
import \* as QSHelpers from "./shared/dashboard-filters-query-stages";  
  
/\*\*  
 \* Abbreviations used for card aliases in this test suite:  
 \* qbq = question-based question  
 \* qbm =...

REQ-666: const { H } = cy;  
  
import \* as QSHelpers from "./shared/dashboard-filters-query-stages";  
  
/\*\*  
 \* Abbreviations used for card aliases in this test suite:  
 \* qbq = question-based question  
 \* qbm =...

REQ-667: const { H } = cy;  
  
import \* as QSHelpers from "./shared/dashboard-filters-query-stages";  
  
/\*\*  
 \* Abbreviations used for card aliases in this test suite:  
 \* qbq = question-based question  
 \* qbm =...

REQ-668: const { H } = cy;  
  
import \* as QSHelpers from "./shared/dashboard-filters-query-stages";  
  
/\*\*  
 \* Abbreviations used for card aliases in this test suite:  
 \* qbq = question-based question  
 \* qbm =...

REQ-669: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import type { Card, StructuredQuery } from "metabase-types/api";  
  
import \* as QSHelpers from...

REQ-670: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import type {  
 Card,  
 ConcreteFieldReference,  
 StructuredQuery,  
} from "metabase-types/api";  
  
const {...

REQ-671: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import type { ValuesQueryType } from "metabase-types/api";  
  
import { matrix } from "./matrix";  
  
export type...

REQ-672: import type { TestCase } from "./matrix-helpers";  
  
export const matrix: TestCase[] = [  
 {  
 arity: "single",  
 type: "search",  
 adminType: "search",  
 operator: "Is",  
 source:...

REQ-673: import { runPage } from "./helpers/matrix-helpers";  
  
describe("scenarios > dashboard > parameters > matrix", () => {  
 runPage(0);  
});

REQ-674: import { runPage } from "./helpers/matrix-helpers";  
  
describe("scenarios > dashboard > parameters > matrix", () => {  
 runPage(1);  
});

REQ-675: import { runPage } from "./helpers/matrix-helpers";  
  
describe("scenarios > dashboard > parameters > matrix", () => {  
 runPage(10);  
});

REQ-676: import { runPage } from "./helpers/matrix-helpers";  
  
describe("scenarios > dashboard > parameters > matrix", () => {  
 runPage(2);  
});

REQ-677: import { runPage } from "./helpers/matrix-helpers";  
  
describe("scenarios > dashboard > parameters > matrix", () => {  
 runPage(3);  
});

REQ-678: import { runPage } from "./helpers/matrix-helpers";  
  
describe("scenarios > dashboard > parameters > matrix", () => {  
 runPage(4);  
});

REQ-679: import { runPage } from "./helpers/matrix-helpers";  
  
describe("scenarios > dashboard > parameters > matrix", () => {  
 runPage(5);  
});

REQ-680: import { runPage } from "./helpers/matrix-helpers";  
  
describe("scenarios > dashboard > parameters > matrix", () => {  
 runPage(6);  
});

REQ-681: import { runPage } from "./helpers/matrix-helpers";  
  
describe("scenarios > dashboard > parameters > matrix", () => {  
 runPage(7);  
});

REQ-682: import { runPage } from "./helpers/matrix-helpers";  
  
describe("scenarios > dashboard > parameters > matrix", () => {  
 runPage(8);  
});

REQ-683: import { runPage } from "./helpers/matrix-helpers";  
  
describe("scenarios > dashboard > parameters > matrix", () => {  
 runPage(9);  
});

REQ-684: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 ORDERS\_BY\_YEAR\_QUESTION\_ID,  
 ORDERS\_DASHBOARD\_ID,  
} from...

REQ-685: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import { ORDERS\_DASHBOARD\_ID } from "e2e/support/cypress\_sample\_instance\_data";  
import { createMockParameter }...

REQ-686: const { H } = cy;  
  
import {  
 guiDashboard,  
 guiQuestion,  
 mapGUIDashboardParameters,  
 mapNativeDashboardParameters,  
 nativeDashboardDetails,  
 nativeQuestionDetails,  
} from...

REQ-687: const { H } = cy;  
  
import \* as SQLFilter from "../native-filters/helpers/e2e-sql-filter-helpers";  
  
import {  
 questionDetails as questionDetails2,  
 questionDetailsWithDefaults,  
} from...

REQ-688: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 ORDERS\_DASHBOARD\_ID,  
 ORDERS\_QUESTION\_ID,  
} from...

REQ-689: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import { ORDERS\_DASHBOARD\_ID } from "e2e/support/cypress\_sample\_instance\_data";  
import { defer } from...

REQ-690: const { H } = cy;  
import { METABASE\_SECRET\_KEY } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 ORDERS\_DASHBOARD\_ID,  
 ...

REQ-691: const { H } = cy;  
import {  
 ORDERS\_DASHBOARD\_ID,  
 ORDERS\_QUESTION\_ID,  
} from "e2e/support/cypress\_sample\_instance\_data";  
  
import { IFRAME\_CODE, getEmbeddingJsCode } from...

REQ-692: const { H } = cy;  
import { USERS, USER\_GROUPS, WRITABLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 ...

REQ-693: import { ORDERS\_DASHBOARD\_ID } from "e2e/support/cypress\_sample\_instance\_data";  
import {  
 mockAuthSsoEndpointForSamlAuthProvider,  
 stubWindowOpenForSamlPopup,  
} from...

REQ-694: import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 ORDERS\_COUNT\_QUESTION\_ID,  
 ORDERS\_DASHBOARD\_ID,  
 ORDERS\_QUESTION\_ID,  
 THIRD\_COLLECTION\_ID,  
} from...

REQ-695: import {  
 FIRST\_COLLECTION\_ID,  
 ORDERS\_DASHBOARD\_ID,  
 ORDERS\_QUESTION\_ID,  
} from "e2e/support/cypress\_sample\_instance\_data";  
  
const { H } = cy;  
  
describe("scenarios > embedding > sdk iframe embed...

REQ-696: const { H } = cy;  
  
describe("scenarios > embedding > sdk iframe embedding > without token features", () => {  
 beforeEach(() => {  
 H.prepareSdkIframeEmbedTest({  
 withTokenFeatures: false,  
  
 ...

REQ-697: import {  
 ORDERS\_COUNT\_QUESTION\_ID,  
 ORDERS\_DASHBOARD\_ENTITY\_ID,  
 ORDERS\_DASHBOARD\_ID,  
 ORDERS\_QUESTION\_ENTITY\_ID,  
 ORDERS\_QUESTION\_ID,  
} from "e2e/support/cypress\_sample\_instance\_data";  
  
const...

REQ-698: import { ORDERS\_DASHBOARD\_ID } from "e2e/support/cypress\_sample\_instance\_data";  
import type { MetabaseTheme } from "metabase/embedding-sdk/theme/MetabaseTheme";  
  
const { H } = cy;  
  
const LIGHT\_THEME...

REQ-699: const { H } = cy;  
  
const suiteTitle =  
 "scenarios > embedding > sdk iframe embed setup > auto enable embedding settings";  
  
describe(suiteTitle, () => {  
 beforeEach(() => {  
 H.restore();  
 ...

REQ-700: import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
import {  
 codeBlock,  
 getEmbedSidebar,  
 navigateToEmbedOptionsStep,  
 navigateToEntitySelectionStep,  
} from...

REQ-701: import {  
 ORDERS\_COUNT\_QUESTION\_ID,  
 ORDERS\_DASHBOARD\_ID,  
} from "e2e/support/cypress\_sample\_instance\_data";  
import { enableJwtAuth } from "e2e/support/helpers/e2e-jwt-helpers";  
import {...

REQ-702: import { match } from "ts-pattern";  
  
import { entityPickerModal } from "e2e/support/helpers";  
import type { Dashboard, RecentItem } from "metabase-types/api";  
  
type RecentActivityIntercept = {  
 ...

REQ-703: import {  
 ORDERS\_BY\_YEAR\_QUESTION\_ID,  
 ORDERS\_COUNT\_QUESTION\_ID,  
 ORDERS\_DASHBOARD\_ID,  
} from "e2e/support/cypress\_sample\_instance\_data";  
import { mockEmbedJsToDevServer } from...

REQ-704: import { ORDERS\_COUNT\_QUESTION\_ID } from "e2e/support/cypress\_sample\_instance\_data";  
  
import {  
 assertDashboard,  
 assertRecentItemName,  
 getEmbedSidebar,  
 visitNewEmbedPage,  
} from...

REQ-705: import { mockEmbedJsToDevServer } from "e2e/support/helpers";  
  
import {  
 codeBlock,  
 getEmbedSidebar,  
 navigateToEmbedOptionsStep,  
} from "./helpers";  
  
const { H } = cy;  
  
const DASHBOARD\_NAME =...

REQ-706: import { enableJwtAuth } from "e2e/support/helpers/e2e-jwt-helpers";  
  
import {  
 codeBlock,  
 getEmbedSidebar,  
 navigateToEmbedOptionsStep,  
 navigateToGetCodeStep,  
} from "./helpers";  
  
const { H }...

REQ-707: import { produce } from "immer";  
  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS, PEOPLE } = SAMPLE\_DATABASE;  
  
export const questionDetails = {  
 native: {  
 ...

REQ-708: import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { PEOPLE, PRODUCTS, PRODUCTS\_ID } = SAMPLE\_DATABASE;  
  
export const nativeQuestionDetails = {  
 name: "Count of People by...

REQ-709: import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS, PEOPLE } = SAMPLE\_DATABASE;  
  
const query = `  
SELECT orders.id, orders.product\_id, orders.created\_at AS...

REQ-710: import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS, ORDERS\_ID, PEOPLE, PEOPLE\_ID } = SAMPLE\_DATABASE;  
  
export const regularQuestion = {  
 name: "Orders4t#7 t3",  
 ...

REQ-711: export const getEmbeddingJsCode = ({  
 type,  
 id,  
 downloads,  
 theme,  
 // Match the actual default value (metabase#43838)  
 background = true,  
}) => {  
 return new RegExp(  
 `// you will need...

REQ-712: const { H } = cy;  
import { USERS } from "e2e/support/cypress\_data";  
import {  
 ORDERS\_DASHBOARD\_DASHCARD\_ID,  
 ORDERS\_QUESTION\_ID,  
} from "e2e/support/cypress\_sample\_instance\_data";  
import {...

REQ-713: import type { Sheet } from "xlsx";  
  
import { WRITABLE\_DB\_ID } from "e2e/support/cypress\_data";  
import type {  
 DashboardDetails,  
 NativeQuestionDetails,  
 StructuredQuestionDetails,  
} from...

REQ-714: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS\_ID, ORDERS, PEOPLE\_ID, PRODUCTS\_ID,...

REQ-715: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { PRODUCTS\_ID, PRODUCTS, ORDERS\_ID, ORDERS } =...

REQ-716: const { H } = cy;  
  
const STRING\_CASES = [  
 {  
 title: "is",  
 columnName: "Category",  
 operator: "Is",  
 options: ["Widget"],  
 expectedDisplayName: "Category is Widget",  
 ...

REQ-717: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS, ORDERS\_ID, PRODUCTS, PRODUCTS\_ID,...

REQ-718: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { PRODUCTS\_ID, PEOPLE\_ID } = SAMPLE\_DATABASE;  
  
describe("operators in questions", () => {  
 ...

REQ-719: import dayjs from "dayjs";  
  
import "metabase/lib/dayjs";  
  
const { H } = cy;  
  
const STARTING\_FROM\_UNITS = [  
 "minutes",  
 "hours",  
 "days",  
 "weeks",  
 "months",  
 "quarters",  
 ...

REQ-720: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { PRODUCTS, PRODUCTS\_ID } =...

REQ-721: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { PRODUCTS } = SAMPLE\_DATABASE;  
  
describe("scenarios > question > view", () => {  
 beforeEach(() => {  
 ...

REQ-722: import dayjs from "dayjs";  
  
const { H } = cy;  
import {  
 SAMPLE\_DB\_ID,  
 SAMPLE\_DB\_SCHEMA\_ID,  
 USER\_GROUPS,  
 WRITABLE\_DB\_ID,  
} from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from...

REQ-723: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { REVIEWS, REVIEWS\_ID } =...

REQ-724: const { H } = cy;  
  
import {  
 SAMPLE\_DB\_ID,  
 SAMPLE\_DB\_SCHEMA\_ID,  
 WRITABLE\_DB\_ID,  
} from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const...

REQ-725: const { H } = cy;  
  
const paths = [  
 "/",  
 "/getting-started",  
 "/collection/root",  
 "/browse/models",  
 "/browse/databases",  
 "/browse/metrics",  
 "/trash",  
 "/admin",  
];  
  
const locales = [  
 ...

REQ-726: import { ORDERS\_MODEL\_ID } from "e2e/support/cypress\_sample\_instance\_data";  
  
const { H } = cy;  
  
type TestCase = {  
 operator: string;  
 lhsExpression: string;  
 rhsExpression: string;  
 ...

REQ-727: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const {  
 ORDERS,  
 ORDERS\_ID,  
 PRODUCTS,  
 ...

REQ-728: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS, ORDERS\_ID, PRODUCTS, PRODUCTS\_ID } =...

REQ-729: # Files used by docker-compose maildev-ssl

REQ-730: const { H } = cy;  
  
const loremIpsum =  
 "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer auctor id erat non sollicitudin. ";  
  
describe("Metabot UI", () => {  
 beforeEach(() => {  
 ...

REQ-731: const { H } = cy;  
import { USERS } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 FIRST\_COLLECTION\_ID,  
 ORDERS\_MODEL\_ID,  
} from...

REQ-732: const { H } = cy;  
import { USER\_GROUPS } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 FIRST\_COLLECTION\_ID,  
 ORDERS\_MODEL\_ID,  
}...

REQ-733: const { H } = cy;  
import { USER\_GROUPS } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 FIRST\_COLLECTION\_ID,  
 ...

REQ-734: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import { ORDERS\_MODEL\_ID } from "e2e/support/cypress\_sample\_instance\_data";  
  
const { ORDERS\_ID, ORDERS,...

REQ-735: const { H } = cy;  
import { USER\_GROUPS } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 FIRST\_COLLECTION\_ID,  
 ORDERS\_MODEL\_ID,  
}...

REQ-736: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS\_ID } = SAMPLE\_DATABASE;  
  
const ORDERS\_SCALAR\_METRIC = {  
 name: "Count of orders",  
 type:...

REQ-737: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import type { StructuredQuestionDetails } from "e2e/support/helpers";  
  
const { ORDERS\_ID, ORDERS } =...

REQ-738: const { H } = cy;  
import { USERS } from "e2e/support/cypress\_data";  
  
describe("scenarios > models > create", () => {  
 beforeEach(() => {  
 H.restore();  
 cy.signInAsAdmin();  
 ...

REQ-739: import {  
 NativeEditor,  
 entityPickerModal,  
 entityPickerModalTab,  
 interceptIfNotPreviouslyDefined,  
 modal,  
 openQuestionActions,  
 popover,  
} from "e2e/support/helpers";  
  
export function...

REQ-740: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import { createModelIndex } from "e2e/support/helpers/e2e-model-index-helper";  
  
const { PRODUCTS\_ID, PEOPLE\_ID...

REQ-741: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
import { startQuestionFromModel } from "./helpers/e2e-models-helpers";  
  
const { PEOPLE, PRODUCTS,...

REQ-742: const { H } = cy;  
import { ORDERS\_QUESTION\_ID } from "e2e/support/cypress\_sample\_instance\_data";  
  
import { selectFromDropdown } from "./helpers/e2e-models-helpers";  
  
describe("scenarios > models...

REQ-743: const { H } = cy;  
import { ORDERS\_BY\_YEAR\_QUESTION\_ID } from "e2e/support/cypress\_sample\_instance\_data";  
  
describe("scenarios > models > revision history", () => {  
 beforeEach(() => {  
 ...

REQ-744: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
import { turnIntoModel } from "./helpers/e2e-models-helpers";  
const { ORDERS, ORDERS\_ID } =...

REQ-745: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 ORDERS\_BY\_YEAR\_QUESTION\_ID,  
 ...

REQ-746: const { H } = cy;  
import { SAMPLE\_DB\_ID, SAMPLE\_DB\_SCHEMA\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 ORDERS\_DASHBOARD\_ID,  
...

REQ-747: const { H } = cy;  
  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 FIRST\_COLLECTION\_ID,  
 ORDERS\_DASHBOARD\_ID,  
 ORDERS\_MODEL\_ID,  
 ORDERS\_QUESTION\_ID,  
} from...

REQ-748: import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
  
const { H } = cy;  
  
const FIX\_MESSAGE = "Fixes applied. Run your query to view results.";  
  
describe.skip("scenarios > native > ai sql fixer",...

REQ-749: const { H } = cy;  
import { SAMPLE\_DB\_ID, USER\_GROUPS } from "e2e/support/cypress\_data";  
  
const PG\_DB\_ID = 2;  
const mongoName = "QA Mongo";  
const postgresName = "QA Postgres12";  
const additionalPG =...

REQ-750: const { H } = cy;  
import { SAMPLE\_DB\_ID, USER\_GROUPS } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
import {  
 getRunQueryButton,  
 ...

REQ-751: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import type {  
 NativeQuestionDetails,  
 ...

REQ-752: const { H } = cy;  
import {  
 SAMPLE\_DB\_ID,  
 USER\_GROUPS,  
 WRITABLE\_DB\_ID,  
} from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {...

REQ-753: const { H } = cy;  
import {  
 ADMIN\_PERSONAL\_COLLECTION\_ID,  
 ORDERS\_QUESTION\_ID,  
} from "e2e/support/cypress\_sample\_instance\_data";  
  
describe("scenarios > question > native subquery", () => {  
 ...

REQ-754: const { H } = cy;  
import { USER\_GROUPS } from "e2e/support/cypress\_data";  
  
const { ALL\_USERS\_GROUP } = USER\_GROUPS;  
  
// HACK which lets us type (even very long words) without losing focus  
// this is...

REQ-755: const { H } = cy;  
  
describe("scenarios > question > native > suggestions", () => {  
 beforeEach(() => {  
 H.restore();  
 cy.signInAsNormalUser();  
 });  
  
 it("should show suggestions for tables",...

REQ-756: import { popover, selectDropdown } from "e2e/support/helpers";  
  
const currentYearString = new Date().getFullYear().toString();  
  
export function setMonthAndYear({ month, year } = {}) {  
 ...

REQ-757: import dayjs from "dayjs";  
  
export const STRING\_FILTER\_SUBTYPES = {  
 String: {  
 searchTerm: "Synerg",  
 value: "Synergistic Granite Chair",  
 representativeResult: "Synergistic Granite...

REQ-758: import { filterWidget, popover, selectDropdown } from "e2e/support/helpers";  
  
// FILTER WIDGET TYPE  
  
/\*\*  
 \* Sets a field filter widget type. Depends on the field that field filter is mapped to.  
 \*  
 \*...

REQ-759: import {  
 NativeEditor,  
 filterWidget,  
 selectDropdown,  
} from "e2e/support/helpers";  
  
// FILTER TYPES  
  
/\*\*  
 \* Opens popover with the list of possible SQL filter types to choose from.  
 \* It does...

REQ-760: const { H } = cy;  
  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import type { NativeQuestionDetails } from "e2e/support/helpers";  
import type { GetFieldValuesResponse } from...

REQ-761: const { H } = cy;  
import { SAMPLE\_DB\_ID, USER\_GROUPS } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
import \* as FieldFilter from...

REQ-762: const { H } = cy;  
  
import { WRITABLE\_DB\_ID } from "e2e/support/cypress\_data";  
  
import \* as DateFilter from "./helpers/e2e-date-filter-helpers";  
import {  
 DATE\_FILTER\_SUBTYPES,  
 ...

REQ-763: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
import \* as FieldFilter from "./helpers/e2e-field-filter-helpers";  
import \* as SQLFilter from...

REQ-764: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import type { TemplateTag } from "metabase-types/api";  
  
type SectionId =  
 | "no\_default\_non\_required"  
 |...

REQ-765: const { H } = cy;  
import { SAMPLE\_DB\_ID, USER\_GROUPS } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
import \* as FieldFilter from...

REQ-766: const { H } = cy;  
  
import \* as DateFilter from "./helpers/e2e-date-filter-helpers";  
import \* as SQLFilter from "./helpers/e2e-sql-filter-helpers";  
  
describe("scenarios > filters > sql filters > basic...

REQ-767: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 ORDERS\_DASHBOARD\_ID,  
 THIRD\_COLLECTION\_ID,  
} from...

REQ-768: const { H } = cy;  
  
describe("scenarios > about Metabase", () => {  
 beforeEach(() => {  
 H.restore();  
 cy.signInAsAdmin();  
  
 cy.visit("/");  
 cy.icon("gear").click();  
 //...

REQ-769: import { SAMPLE\_DB\_ID, USER\_GROUPS } from "e2e/support/cypress\_data";  
import {  
 FIRST\_COLLECTION\_ID,  
 SECOND\_COLLECTION\_ID,  
} from "e2e/support/cypress\_sample\_instance\_data";  
  
const { H } =...

REQ-770: const { H } = cy;  
import { USERS } from "e2e/support/cypress\_data";  
  
const { admin } = USERS;  
  
describe("scenarios > auth > password", { tags: "@external" }, () => {  
 beforeEach(() => {  
 ...

REQ-771: const { H } = cy;  
import { USERS } from "e2e/support/cypress\_data";  
  
const sizes = [  
 [1280, 800],  
 [640, 360],  
];  
const { admin } = USERS;  
  
describe("scenarios > auth > signin", () => {  
 ...

REQ-772: const { H } = cy;  
import { USERS } from "e2e/support/cypress\_data";  
  
const { admin } = USERS;  
  
describe("scenarios > auth > signin > SSO", () => {  
 beforeEach(() => {  
 H.restore();  
 ...

REQ-773: const { H } = cy;  
import { SAMPLE\_DB\_ID, USERS } from "e2e/support/cypress\_data";  
import {  
 ADMIN\_PERSONAL\_COLLECTION\_ID,  
 ORDERS\_BY\_YEAR\_QUESTION\_ID,  
 ORDERS\_COUNT\_QUESTION\_ID,  
 ...

REQ-774: const { H } = cy;  
  
H.describeWithSnowplow(  
 "scenarios > embedding-homepage > snowplow events",  
 () => {  
 beforeEach(() => {  
 H.restore("default");  
 H.resetSnowplow();  
  
 ...

REQ-775: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 ORDERS\_MODEL\_ID,  
 ORDERS\_QUESTION\_ID,  
}...

REQ-776: const { H } = cy;  
import { USERS } from "e2e/support/cypress\_data";  
import {  
 ADMIN\_PERSONAL\_COLLECTION\_ID,  
 ORDERS\_BY\_YEAR\_QUESTION\_ID,  
 ORDERS\_DASHBOARD\_ID,  
} from...

REQ-777: const { H } = cy;  
  
describe("metabase > scenarios > navbar > new menu", () => {  
 beforeEach(() => {  
 H.restore();  
 cy.signInAsAdmin();  
  
 cy.visit("/");  
 // eslint-disable-next-line...

REQ-778: const { H } = cy;  
import {  
 createMockVersionInfo,  
 createMockVersionInfoRecord as mockVersion,  
} from "metabase-types/api/mocks";  
  
describe("nav > what's new notification", () => {  
 beforeEach(()...

REQ-779: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS\_ID } = SAMPLE\_DATABASE;  
  
const getQuestionDetails = () => ({  
 name: "Question",  
 query: {  
 ...

REQ-780: const { H } = cy;  
import type { ChecklistItemValue } from "metabase/home/components/Onboarding/types";  
  
describe("Onboarding checklist page", () => {  
 beforeEach(() => {  
 H.restore();  
 ...

REQ-781: const { H } = cy;  
  
describe("scenarios > reference > databases", () => {  
 beforeEach(() => {  
 H.restore();  
 cy.signInAsAdmin();  
 });  
  
 it("should see the listing", () => {  
 ...

REQ-782: const { H } = cy;  
  
describe("issue 5276", () => {  
 beforeEach(() => {  
 H.restore();  
 cy.signInAsAdmin();  
 cy.intercept("PUT", "/api/field/\*").as("updateField");  
 });  
  
 it("should allow...

REQ-783: const { H } = cy;  
  
H.describeWithSnowplowEE("scenarios > setup embedding (EMB-477)", () => {  
 beforeEach(() => {  
 H.resetSnowplow();  
 H.restore("blank");  
 });  
  
 it("should redirect correctly...

REQ-784: const { H } = cy;  
const { IS\_ENTERPRISE } = Cypress.env();  
import { USERS } from "e2e/support/cypress\_data";  
import { SUBSCRIBE\_URL } from "metabase/setup/constants";  
  
const { admin } = USERS;  
  
//...

REQ-785: const { H } = cy;  
import { USERS } from "e2e/support/cypress\_data";  
import { NORMAL\_USER\_ID } from "e2e/support/cypress\_sample\_instance\_data";  
  
const { normal } = USERS;  
  
const { first\_name,...

REQ-786: const { H } = cy;  
import { SAMPLE\_DB\_ID, USERS } from "e2e/support/cypress\_data";  
import {  
 ADMIN\_PERSONAL\_COLLECTION\_ID,  
 FIRST\_COLLECTION\_ID,  
 NORMAL\_PERSONAL\_COLLECTION\_ID,  
 ...

REQ-787: const { H } = cy;  
import { SAMPLE\_DB\_TABLES, USERS } from "e2e/support/cypress\_data";  
import {  
 ADMIN\_PERSONAL\_COLLECTION\_ID,  
 FIRST\_COLLECTION\_ID,  
} from...

REQ-788: const { H } = cy;  
import { ORDERS\_DASHBOARD\_ID } from "e2e/support/cypress\_sample\_instance\_data";  
  
describe("scenarios > dashboard > bookmarks", () => {  
 beforeEach(() => {  
 H.restore();  
 ...

REQ-789: const { H } = cy;  
import { ORDERS\_QUESTION\_ID } from "e2e/support/cypress\_sample\_instance\_data";  
  
import { toggleQuestionBookmarkStatus } from "./helpers/bookmark-helpers";  
  
describe("scenarios >...

REQ-790: const { H } = cy;  
import {  
 ORDERS\_COUNT\_QUESTION\_ID,  
 ORDERS\_QUESTION\_ID,  
} from "e2e/support/cypress\_sample\_instance\_data";  
  
import {  
 createAndBookmarkQuestion,  
 moveBookmark,  
 ...

REQ-791: const { H } = cy;  
import {  
 ORDERS\_COUNT\_QUESTION\_ID,  
 ORDERS\_DASHBOARD\_ID,  
} from "e2e/support/cypress\_sample\_instance\_data";  
  
describe(  
 "scenarios > premium > content verification",  
 { tags:...

REQ-792: const { H } = cy;  
import { USERS } from "e2e/support/cypress\_data";  
import {  
 ORDERS\_DASHBOARD\_ID,  
 ORDERS\_QUESTION\_ID,  
} from "e2e/support/cypress\_sample\_instance\_data";  
  
describe("scenarios >...

REQ-793: const { H } = cy;  
import { USER\_GROUPS, WRITABLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 ...

REQ-794: import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 createQuestion,  
 moveDnDKitElement,  
 navigationSidebar,  
}...

REQ-795: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS, ORDERS\_ID } = SAMPLE\_DATABASE;  
  
const COLLECTION\_NAME = "Official Collection Test";  
  
const...

REQ-796: const { H } = cy;  
import { USERS } from "e2e/support/cypress\_data";  
import {  
 ADMIN\_PERSONAL\_COLLECTION\_ID,  
 FIRST\_COLLECTION\_ID,  
} from "e2e/support/cypress\_sample\_instance\_data";  
  
const { admin }...

REQ-797: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import { ORDERS\_BY\_YEAR\_QUESTION\_ID } from...

REQ-798: const { H } = cy;  
import { SAMPLE\_DB\_ID, USER\_GROUPS } from "e2e/support/cypress\_data";  
import {  
 ORDERS\_DASHBOARD\_ID,  
 ORDERS\_QUESTION\_ID,  
} from "e2e/support/cypress\_sample\_instance\_data";  
  
const...

REQ-799: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 ORDERS\_DASHBOARD\_ID,  
 ORDERS\_QUESTION\_ID,  
} from...

REQ-800: const { H } = cy;  
import { USER\_GROUPS } from "e2e/support/cypress\_data";  
  
const { ALL\_USERS\_GROUP } = USER\_GROUPS;  
  
const NATIVE\_QUERIES\_PERMISSION\_INDEX = 0;  
  
describe("scenarios > admin >...

REQ-801: const { H } = cy;  
import { SAMPLE\_DB\_ID, SAMPLE\_DB\_SCHEMA\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS, ORDERS\_ID } =...

REQ-802: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
  
const DETAILS\_PERMISSION\_INDEX = 4;  
  
describe("scenarios > admin > permissions > database details permissions", () => {  
 ...

REQ-803: const { H } = cy;  
import { USER\_GROUPS } from "e2e/support/cypress\_data";  
  
const { ALL\_USERS\_GROUP } = USER\_GROUPS;  
  
const EE\_DATA\_ACCESS\_PERMISSION\_INDEX = 0;  
const...

REQ-804: const { H } = cy;  
import { SAMPLE\_DB\_ID, USER\_GROUPS } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 ORDERS\_DASHBOARD\_ID,  
 ...

REQ-805: const { H } = cy;  
import { USER\_GROUPS } from "e2e/support/cypress\_data";  
import \* as PH from "e2e/test/scenarios/admin/performance/helpers/e2e-strategy-form-helpers";  
  
const { ALL\_USERS\_GROUP,...

REQ-806: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import {  
 ADMIN\_PERSONAL\_COLLECTION\_ID,  
 ORDERS\_DASHBOARD\_ID,  
 ORDERS\_QUESTION\_ID,  
} from...

REQ-807: const { H } = cy;  
import { SAMPLE\_DB\_ID, USERS, USER\_GROUPS } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 NODATA\_USER\_ID,  
 ...

REQ-808: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import type {  
 ConcreteFieldReference,  
 ...

REQ-809: import { SAMPLE\_DB\_ID, USER\_GROUPS } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import type { StructuredQuestionDetails } from...

REQ-810: import type { Dataset } from "metabase-types/api";  
  
export type DatasetResponse = {  
 body: Dataset;  
 url: string;  
 headers: any;  
 statusCode: number;  
};  
  
export type DashcardQueryResponse = {  
 ...

REQ-811: import { USER\_GROUPS, WRITABLE\_DB\_ID } from "e2e/support/cypress\_data";  
  
import {  
 assertResponseFailsClosed,  
 assignAttributeToUser,  
 configureSandboxPolicy,  
 getCardResponses,  
 gizmoViewer,  
 ...

REQ-812: const { H } = cy;  
import { SAMPLE\_DB\_ID, USERS, USER\_GROUPS } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 NORMAL\_USER\_ID,  
 ...

REQ-813: import { USER\_GROUPS } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import { checkNotNull } from "metabase/lib/types";  
import type {...

REQ-814: const { H } = cy;  
import { SAMPLE\_DB\_ID, USER\_GROUPS } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import { ORDERS\_QUESTION\_ID } from...

REQ-815: const { H } = cy;  
import { ORDERS\_QUESTION\_ID } from "e2e/support/cypress\_sample\_instance\_data";  
  
import { interceptPerformanceRoutes } from...

REQ-816: import \_ from "underscore";  
  
const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import type {...

REQ-817: const { H } = cy;  
  
const PG\_DB\_ID = 2;  
  
describe(  
 "question loading changes document title",  
 { tags: "@external" },  
 () => {  
 beforeEach(() => {  
 H.restore("postgres-12");  
 ...

REQ-818: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import type {  
 DashboardDetails,  
 StructuredQuestionDetails,  
} from "e2e/support/helpers";  
  
const {...

REQ-819: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import type { NativeQuestionDetails } from "e2e/support/helpers";  
  
const ordersTableQuestionDetails: NativeQuestionDetails =...

REQ-820: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS, ORDERS\_ID, PRODUCTS, PRODUCTS\_ID,...

REQ-821: const { H } = cy;  
import { SAMPLE\_DB\_ID, USERS } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 ORDERS\_COUNT\_QUESTION\_ID,  
 ...

REQ-822: const { H } = cy;  
import { WRITABLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 ORDERS\_COUNT\_QUESTION\_ID,  
 ...

REQ-823: const { H } = cy;  
import { SAMPLE\_DB\_ID, USERS } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 ADMIN\_PERSONAL\_COLLECTION\_ID,  
 ...

REQ-824: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 ORDERS\_COUNT\_QUESTION\_ID,  
 ...

REQ-825: const { H } = cy;  
import { SAMPLE\_DB\_ID, WRITABLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import { ADMIN\_USER\_ID } from...

REQ-826: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS, ORDERS\_ID } = SAMPLE\_DATABASE;  
  
describe("scenarios > question > null", () => {  
 ...

REQ-827: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import type { StructuredQuestionDetails } from "e2e/support/helpers";  
import { uuid } from...

REQ-828: const { H } = cy;  
import { WRITABLE\_DB\_ID } from "e2e/support/cypress\_data";  
  
const supportedDatabases = [  
 {  
 database: "Mongo",  
 snapshotName: "mongo-5",  
 dbName: "QA Mongo",  
 },  
 {  
 ...

REQ-829: const { H } = cy;  
  
H.describeWithSnowplow("scenarios > question > snowplow", () => {  
 describe("chart\_generated", () => {  
 const generateNonTableVisualization = () => {  
 cy.visit("/");  
 ...

REQ-830: import { onlyOn } from "@cypress/skip-test";  
  
import { USERS, USER\_GROUPS } from "e2e/support/cypress\_data";  
import {  
 ORDERS\_COUNT\_QUESTION\_ID,  
 ORDERS\_DASHBOARD\_ID,  
 ORDERS\_QUESTION\_ID,  
} from...

REQ-831: const { H } = cy;  
  
import {  
 ORDERS\_QUESTION\_ENTITY\_ID,  
 ORDERS\_QUESTION\_ID,  
} from "e2e/support/cypress\_sample\_instance\_data";  
  
describe("scenarios > questions > entity id support", () => {  
 ...

REQ-832: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 ORDERS\_BY\_YEAR\_QUESTION\_ID,  
 ORDERS\_COUNT\_QUESTION\_ID,  
 ORDERS\_DASHBOARD\_ID,  
 ...

REQ-833: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS, ORDERS\_ID, PRODUCTS, PRODUCTS\_ID } =...

REQ-834: const { H } = cy;  
import { dedent } from "ts-dedent";  
  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import { ORDERS\_QUESTION\_ID } from...

REQ-835: const { H } = cy;  
import { SAMPLE\_DB\_ID, WRITABLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import { ORDERS\_QUESTION\_ID } from...

REQ-836: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import { ORDERS\_QUESTION\_ID } from...

REQ-837: const { H } = cy;  
import {  
 SAMPLE\_DB\_ID,  
 SAMPLE\_DB\_SCHEMA\_ID,  
 USER\_GROUPS,  
 WRITABLE\_DB\_ID,  
} from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from...

REQ-838: const { H } = cy;  
  
import { WRITABLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import { ORDERS\_QUESTION\_ID } from...

REQ-839: const { H } = cy;  
import {  
 ORDERS\_DASHBOARD\_ID,  
 ORDERS\_QUESTION\_ID,  
} from "e2e/support/cypress\_sample\_instance\_data";  
  
describe("search > recently viewed", () => {  
 beforeEach(() => {  
 ...

REQ-840: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 ADMIN\_USER\_ID,  
 FIRST\_COLLECTION\_ID,  
 ...

REQ-841: import \_ from "underscore";  
  
const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS, ORDERS\_ID } = SAMPLE\_DATABASE;  
  
const PAGE\_SIZE = 50;  
const...

REQ-842: const { H } = cy;  
  
import { commandPaletteInput } from "../../../support/helpers/e2e-command-palette-helpers";  
  
H.describeWithSnowplow("scenarios > search > snowplow", () => {  
 const...

REQ-843: const { H } = cy;  
import { USERS } from "e2e/support/cypress\_data";  
  
["admin", "normal"].forEach((user) => {  
 describe(`search > ${user} user`, () => {  
 beforeEach(() => {  
 H.restore();  
 ...

REQ-844: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 ORDERS\_DASHBOARD\_ID,  
 ORDERS\_QUESTION\_ID,  
} from...

REQ-845: const { H } = cy;  
import { USERS } from "e2e/support/cypress\_data";  
import {  
 ORDERS\_BY\_YEAR\_QUESTION\_ID,  
 ORDERS\_COUNT\_QUESTION\_ID,  
 ORDERS\_QUESTION\_ID,  
} from...

REQ-846: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 ORDERS\_BY\_YEAR\_QUESTION\_ID,  
 ORDERS\_QUESTION\_ID,  
} from...

REQ-847: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 ORDERS\_DASHBOARD\_ID,  
 ORDERS\_MODEL\_ID,  
 ORDERS\_QUESTION\_ID,  
} from...

REQ-848: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import { ORDERS\_QUESTION\_ID } from "e2e/support/cypress\_sample\_instance\_data";  
  
const { PEOPLE\_ID } =...

REQ-849: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 ORDERS\_DASHBOARD\_DASHCARD\_ID,  
 ORDERS\_DASHBOARD\_ID,  
 ORDERS\_QUESTION\_ID,  
} from...

REQ-850: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS, ORDERS\_ID, REVIEWS, REVIEWS\_ID,...

REQ-851: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { PRODUCTS, ORDERS\_ID } = SAMPLE\_DATABASE;  
  
const questionDetails = {  
 name: "sql param",  
 native: {  
...

REQ-852: import xlsx from "xlsx";  
  
const { H } = cy;  
  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { PEOPLE } = SAMPLE\_DATABASE;  
  
const questionData = {  
 name: "Parameterized...

REQ-853: const { H } = cy;  
import {  
 ORDERS\_BY\_YEAR\_QUESTION\_ID,  
 ORDERS\_DASHBOARD\_DASHCARD\_ID,  
 ORDERS\_DASHBOARD\_ID,  
} from "e2e/support/cypress\_sample\_instance\_data";  
  
/\*\* These tests are about the...

REQ-854: const { H } = cy;  
  
["dashboard", "question"].forEach((resource) => {  
 describe(`embed modal behavior for ${resource}s`, () => {  
 beforeEach(() => {  
 H.restore();  
 cy.signInAsAdmin();  
  
 ...

REQ-855: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS\_ID } = SAMPLE\_DATABASE;  
  
const...

REQ-856: const { H } = cy;  
import { SAMPLE\_DB\_ID, USERS, WEBMAIL\_CONFIG } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 ADMIN\_USER\_ID,  
 ...

REQ-857: const { H } = cy;  
import { USERS } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import { ORDERS\_DASHBOARD\_ID } from...

REQ-858: const { H } = cy;  
  
H.describeWithSnowplow("scenarios > stats > snowplow", () => {  
 beforeEach(() => {  
 H.restore();  
 H.resetSnowplow();  
 cy.signInAsAdmin();  
 H.enableTracking();  
 });  
  
 ...

REQ-859: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS, ORDERS\_ID, PEOPLE, PRODUCTS,...

REQ-860: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { PRODUCTS, PRODUCTS\_ID, ORDERS\_ID, ORDERS } =...

REQ-861: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { PEOPLE\_ID, PEOPLE } =...

REQ-862: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS\_ID } = SAMPLE\_DATABASE;  
  
describe("scenarios > visualizations > gauge chart", () => {  
 ...

REQ-863: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS, ORDERS\_ID, PEOPLE, PRODUCTS } = SAMPLE\_DATABASE;  
  
const ORDERS\_CREATED\_AT\_FIELD\_REF = [  
 ...

REQ-864: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS, ORDERS\_ID, PRODUCTS } =...

REQ-865: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS, ORDERS\_ID, PRODUCTS, PRODUCTS\_ID,...

REQ-866: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { PEOPLE, PEOPLE\_ID } =...

REQ-867: const { H } = cy;  
  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const {  
 ACCOUNTS,  
 ACCOUNTS\_ID,  
 PRODUCTS,  
 ...

REQ-868: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS\_ID } = SAMPLE\_DATABASE;  
  
describe("scenarios > visualizations > progress chart", () => {  
 ...

REQ-869: const { H } = cy;  
  
describe("scenarios > visualizations > rows", () => {  
 beforeEach(() => {  
 H.restore();  
 cy.signInAsAdmin();  
 });  
  
 // Until we enable multi-browser support, this repro...

REQ-870: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
  
const SANKEY\_QUERY = `  
SELECT 'Social Media' AS source, 'Landing Page' AS target, 30000 AS metric  
UNION ALL  
SELECT 'Email...

REQ-871: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS, ORDERS\_ID, PRODUCTS } =...

REQ-872: const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS\_ID, ORDERS, PRODUCTS\_ID, PRODUCTS } = SAMPLE\_DATABASE;  
  
describe("scenarios > question >...

REQ-873: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS, ORDERS\_ID } = SAMPLE\_DATABASE;  
const...

REQ-874: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import type { StructuredQuestionDetails } from...

REQ-875: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS, ORDERS\_ID, PRODUCTS } =...

REQ-876: import \_ from "underscore";  
  
const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { PEOPLE,...

REQ-877: const { H } = cy;  
import { SAMPLE\_DB\_ID, USER\_GROUPS } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS, ORDERS\_ID, PRODUCTS,...

REQ-878: import \_ from "underscore";  
  
const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {...

REQ-879: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { PEOPLE, PEOPLE\_ID } =...

REQ-880: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import { ORDERS\_COUNT\_QUESTION\_ID } from...

REQ-881: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const {  
 ORDERS,  
 ORDERS\_ID,  
 PRODUCTS,  
 ...

REQ-882: const { H } = cy;  
import { SAMPLE\_DB\_ID, WRITABLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const {  
 ORDERS,  
 ORDERS\_ID,  
 ...

REQ-883: const { H } = cy;  
import { SAMPLE\_DB\_ID, USER\_GROUPS } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import { PIVOT\_TABLE\_BODY\_LABEL } from...

REQ-884: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS, ORDERS\_ID } =...

REQ-885: import Color from "color";  
  
const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import { colors } from "metabase/lib/colors";  
  
const { ORDERS, ORDERS\_ID } =...

REQ-886: import \_ from "underscore";  
  
const { H } = cy;  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
  
const { ORDERS\_ID, ORDERS, PRODUCTS\_ID, PRODUCTS } = SAMPLE\_DATABASE;  
  
const...

REQ-887: const { H } = cy;  
import { SAMPLE\_DB\_ID, WRITABLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import {  
 ...

REQ-888: const { H } = cy;  
import { SAMPLE\_DB\_ID } from "e2e/support/cypress\_data";  
import { SAMPLE\_DATABASE } from "e2e/support/cypress\_sample\_database";  
import { ADMIN\_USER\_ID } from...

REQ-889: import {  
 mockAuthProviderAndJwtSignIn,  
 signInAsAdminAndEnableEmbeddingSdk,  
} from "e2e/support/helpers/embedding-sdk-testing";  
  
const TIMEOUT\_MS = 40000;  
  
const CLIENT\_PORT =...

REQ-890: !/usr/bin/env node

REQ-891: Usage of files in this directory and its subdirectories, and of Metabase Enterprise Edition features, is subject to  
the Metabase Commercial License (https://www.metabase.com/license/commercial), and...

REQ-892: Metabase Enterprise Edition  
# License

REQ-893: ## EE Code Structure Notes

REQ-894: {  
 "rules": {  
 // Note: adding this rule to a eslint config file in a subfolder will remove  
 // \*not\* carry over the restricted imports from parent folders, you will  
 // need to copy them...

REQ-895: {  
 "rules": {  
 "no-restricted-imports": [  
 "error",  
 {  
 "patterns": [  
 {  
 "group": [  
 "embedding-sdk/\*",  
 "metabase/\*",  
 ...

REQ-896: Common auth functions

REQ-897: import \* as MetabaseError from "embedding-sdk/errors";  
import type { MetabaseAuthMethod } from "embedding-sdk/types";  
  
export async function connectToInstanceAuthSso(  
 url: string,  
 {  
 headers,  
...

REQ-898: export { connectToInstanceAuthSso } from "./connect-to-instance-auth-sso";  
export { jwtDefaultRefreshTokenFunction } from "./jwt";  
export { openSamlLoginPopup } from "./saml";  
export {...

REQ-899: import \* as MetabaseError from "embedding-sdk/errors";  
  
export async function jwtDefaultRefreshTokenFunction(  
 responseUrl: string,  
 instanceUrl: string,  
 requestHeaders: Record<string, string>,  
 ...

REQ-900: import type { MetabaseEmbeddingSessionToken } from "embedding-sdk/types/refresh-token";  
  
class TypedStorage<T> {  
 constructor(private key: string) {}  
  
 // Get data with proper typing  
 get(): T |...

REQ-901: import \* as MetabaseError from "embedding-sdk/errors";  
import type { MetabaseEmbeddingSessionToken } from "embedding-sdk/types/refresh-token";  
  
/\*  
 \* For the markup for the popup (nice rhyme), visit  
...

REQ-902: import \* as MetabaseError from "embedding-sdk/errors";  
  
export function validateSessionToken(session: any) {  
 if (!session || typeof session !== "object") {  
 throw...

REQ-903: export { DataSourceSelector } from "./DataSelector";

REQ-904: import userEvent from "@testing-library/user-event";  
  
import { createMockMetadata } from "\_\_support\_\_/metadata";  
import { getIcon, render, screen } from "\_\_support\_\_/ui";  
import { delay } from...

REQ-905: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-906: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-907: import { createMockEntitiesState } from "\_\_support\_\_/store";  
import { render, renderWithProviders, screen } from "\_\_support\_\_/ui";  
import { checkNotNull } from "metabase/lib/types";  
import {...

REQ-908: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-909: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-910: import { render, screen } from "\_\_support\_\_/ui";  
  
import DataSelectorSchemaPicker from "./DataSelectorSchemaPicker";  
  
describe("DataSelectorSchemaPicker", () => {  
 it("displays schema name", () =>...

REQ-911: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-912: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-913: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-914: export { SimpleDataPicker } from "./SimpleDataPicker";  
export { SimpleDataPickerView } from "./SimpleDataPickerView";

REQ-915: import { t } from "ttag";  
  
import type { DataPickerDataType, DataTypeInfoItem } from "./types";  
  
export const CONTAINER\_WIDTH = 300;  
  
type DataBucket = "MODELS" | "RAW\_DATA" |...

REQ-916: import { t } from "ttag";  
  
export const CARD\_INFO = {  
 question: {  
 get title() {  
 return t`Saved Questions`;  
 },  
 model: "card",  
 icon: "table2",  
 },  
 model: {  
 get title() {  
...

REQ-917: export const findCollectionById = (collections, collectionId) => {  
 if (!collections || collections.length === 0) {  
 return null;  
 }  
  
 const collection = collections.find((c) => c.id ===...

REQ-918: import type { IconName } from "metabase/ui";  
  
export type DataPickerDataType = "models" | "raw-data" | "questions";  
  
export type DataTypeInfoItem = {  
 id: DataPickerDataType;  
 icon: IconName;  
 ...

REQ-919: import {  
 MODELS\_INFO\_ITEM,  
 RAW\_DATA\_INFO\_ITEM,  
 SAVED\_QUESTIONS\_INFO\_ITEM,  
} from "./constants";  
import type { DataTypeInfoItem } from "./types";  
  
export function getDataTypes({  
 hasModels,  
 ...

REQ-920: export type SdkLoaderData = {  
 className: string;  
 size?: string;  
 color?: string;  
};  
  
export const getSdkLoaderCss = ({  
 className,  
 size = "1.5rem",  
 // eslint-disable-next-line...

REQ-921: eslint-disable no-undef,import/no-commonjs

REQ-922: # [0.55.5-metabot](https://github.com/metabase/metabase/compare/embedding-sdk-0.55.4-metabot...embedding-sdk-0.55.5-metabot) (2025-05-07)  
## Bug Fixes

REQ-923: Metabase embedded analytics SDK

REQ-924: {  
 "rules": {  
 "no-literal-metabase-strings": "off",  
 "no-console": "off",  
 },  
}

REQ-925: !/usr/bin/env node

REQ-926: import fs from "fs";  
import path from "path";  
  
import { getPublicComponents } from "embedding-sdk/bin/get-public-components";  
  
const destinationDir = path.resolve(  
 \_\_dirname,  
 ...

REQ-927: import { type ExportSpecifier, type Node, Project, SyntaxKind } from "ts-morph";  
  
export type ComponentDefinition = {  
 mainComponent: string;  
};  
  
export function getPublicComponents() {  
 const...

REQ-928: import { EMBEDDING\_SDK\_CONFIG } from "metabase/embedding-sdk/config";  
  
// Enable SDK mode as we are in the SDK bundle  
// This applies to SDK derivatives such as new iframe...

REQ-929: {  
 "rules": {  
 "no-literal-metabase-strings": "off",  
 "no-color-literals": "off",  
 "no-console": "off"  
 }  
}

REQ-930: import { runCli } from "../run";  
import { printError } from "../utils/print";  
  
export async function start() {  
 // When the user runs the CLI with npx, there will be some deprecation warnings that...

REQ-931: import { Command } from "commander";  
  
import { start } from "./actions/start";  
  
const program = new Command();  
  
program  
 .name("metabase-embedding-sdk-cli")  
 .description("Metabase Embedding SDK...

REQ-932: export const IMAGE\_NAME = "metabase/metabase-enterprise:latest";  
export const CONTAINER\_NAME = "metabase-enterprise-embedding";  
export const SITE\_NAME = "Metabase Embedding SDK Demo";  
export const...

REQ-933: \* Show popular database engines in the CLI

REQ-934: import { SITE\_NAME } from "./config";  
  
/\*\*  
 \* Use the same setup token for every demo instance.  
 \* This makes it easy to configure across runs.  
 \*/  
export const EMBEDDING\_DEMO\_SETUP\_TOKEN =  
 ...

REQ-935: import { SANDBOXED\_GROUP\_NAMES } from "./config";  
  
const [GROUP\_A, GROUP\_B, GROUP\_C] = SANDBOXED\_GROUP\_NAMES;  
  
/\*\*  
 \* Sample hardcoded users for JWT authentication.  
 \*/  
export const HARDCODED\_USERS =...

REQ-936: import { blue, green, yellow } from "chalk";  
  
import {  
 CONTAINER\_NAME,  
 SAMPLE\_CREDENTIALS\_FILE\_NAME,  
 SDK\_DOCS\_LINK,  
} from "./config";  
  
export const SHOW\_ON\_STARTUP\_MESSAGE = `  
 This tool will...

REQ-937: export const MOCK\_SERVER\_PACKAGE\_JSON = {  
 name: "mock-server",  
 version: "1.0.0",  
 main: "server.js",  
 scripts: {  
 start: "node server.js",  
 },  
 license: "MIT",  
 dependencies: {  
 cors:...

REQ-938: import {  
 addDatabaseConnectionStep,  
 askForTenancyColumns,  
 askIfHasDatabase,  
 checkIfDockerContainerExists,  
 checkIfReactProject,  
 checkIsDockerRunning,  
 checkSdkAvailable,  
 createApiKey,  
 ...

REQ-939: export const ANALYTICS\_CSS\_SNIPPET = `  
body {  
 margin: 0;  
}  
  
.theme-switcher {  
 width: 28px;  
 height: 28px;  
 cursor: pointer;  
}  
  
.analytics-root {  
 font-family:...

REQ-940: import { SDK\_PACKAGE\_NAME } from "../constants/config";  
import type { DashboardInfo } from "../types/dashboard";  
  
interface Options {  
 dashboards: DashboardInfo[];  
 userSwitcherEnabled:...

REQ-941: \*

REQ-942: export const ANALYTICS\_PROVIDER\_SNIPPET\_MINIMAL = `  
import {createContext, useState} from 'react'  
  
/\*\*  
 \* @typedef {Object} AnalyticsContextType  
 \* @property {'light'|'dark'} themeKey - The current...

REQ-943: import { SDK\_PACKAGE\_NAME } from "../constants/config";  
  
interface Options {  
 instanceUrl: string;  
 apiKey: string;  
 userSwitcherEnabled: boolean;  
}  
  
export const getEmbeddingProviderSnippet =...

REQ-944: import { sample } from "underscore";  
  
import { HARDCODED\_JWT\_SHARED\_SECRET } from "../constants/config";  
import { HARDCODED\_USERS } from "../constants/hardcoded-users";  
import type { CliState } from...

REQ-945: import type { DashboardInfo } from "../types/dashboard";  
import { withNextJsUseClientDirective } from "../utils/nextjs-helpers";  
  
import { getAnalyticsDashboardSnippet } from...

REQ-946: export \* from "./analytics-provider-snippet";  
export \* from "./embedding-provider-snippet";  
export \* from "./analytics-dashboard-snippet";  
export \* from "./analytics-page-snippet";  
export \* from...

REQ-947: \*

REQ-948: eslint-disable-next-line no-unconditional-metabase-links-render -- cli snippets

REQ-949: import { HARDCODED\_USERS } from "../constants/hardcoded-users";  
  
export const getUserSwitcherSnippet = () => {  
 const users = HARDCODED\_USERS.map((user) => ({  
 email: user.email,  
 firstName:...

REQ-950: import { search } from "@inquirer/prompts";  
import ora from "ora";  
  
import type { CliStepMethod } from "embedding-sdk/cli/types/cli";  
import type { Settings } from...

REQ-951: import toggle from "inquirer-toggle";  
  
import type { CliStepMethod } from "embedding-sdk/cli/types/cli";  
import { printHelperText } from "embedding-sdk/cli/utils/print";  
  
/\*\*  
 \* Asks the user first...

REQ-952: import { search } from "@inquirer/prompts";  
import toggle from "inquirer-toggle";  
  
import type { CliStepMethod } from "../types/cli";  
import { printHelperText } from "../utils/print";  
  
export const...

REQ-953: import { exec } from "child\_process";  
  
import toggle from "inquirer-toggle";  
import { promisify } from "util";  
  
import { CONTAINER\_NAME } from "../constants/config";  
import {...

REQ-954: import { exec as execCallback } from "child\_process";  
  
import ora from "ora";  
import { promisify } from "util";  
  
import type { CliError, CliStepMethod } from "embedding-sdk/cli/types/cli";  
  
const...

REQ-955: import ora from "ora";  
import semver from "semver";  
  
import {  
 MISSING\_REACT\_DEPENDENCY,  
 PACKAGE\_JSON\_NOT\_FOUND\_MESSAGE,  
 UNSUPPORTED\_REACT\_VERSION,  
} from...

REQ-956: import ora from "ora";  
  
import { SDK\_PACKAGE\_NAME } from "../constants/config";  
import { installSdk } from "../steps/install-sdk";  
import type { CliStepMethod } from "../types/cli";  
import {...

REQ-957: import ora from "ora";  
  
import type { CliStepMethod } from "embedding-sdk/cli/types/cli";  
  
export const createApiKey: CliStepMethod = async (state) => {  
 if (!state.instanceUrl || !state.cookie) {  
 ...

REQ-958: import ora from "ora";  
  
import { createCollection } from "embedding-sdk/cli/utils/create-collection";  
  
import type { CliStepMethod } from "../types/cli";  
import type { DashboardInfo } from...

REQ-959: import fs from "fs/promises";  
  
import { input } from "@inquirer/prompts";  
  
import { getGeneratedComponentFilesMessage } from "../constants/messages";  
import { ANALYTICS\_CSS\_SNIPPET } from...

REQ-960: import fs from "fs/promises";  
  
import { SAMPLE\_CREDENTIALS\_FILE\_NAME } from "../constants/config";  
import type { CliStepMethod } from "../types/cli";  
import { addFileToGitIgnore } from...

REQ-961: import { input } from "@inquirer/prompts";  
  
import { isEmail } from "metabase/lib/email";  
  
import type { CliStepMethod } from "../types/cli";  
import { generateRandomDemoPassword } from...

REQ-962: import fs from "fs/promises";  
  
import { input } from "@inquirer/prompts";  
  
import { installMockServerDeps } from "embedding-sdk/cli/utils/install-mock-server-deps";  
  
import { MOCK\_SERVER\_PACKAGE\_JSON...

REQ-963: export \* from "./check-docker-running";  
export \* from "./check-docker-container-exist";  
export \* from "./create-api-key";  
export \* from "./generate-credentials";  
export \* from...

REQ-964: import { exec as execCallback } from "child\_process";  
  
import { detect } from "detect-package-manager";  
import toggle from "inquirer-toggle";  
import ora from "ora";  
import { promisify } from...

REQ-965: import { checkbox } from "@inquirer/prompts";  
import ora from "ora";  
  
import type { CliStepMethod } from "embedding-sdk/cli/types/cli";  
import type { Table, TableId } from...

REQ-966: import ora from "ora";  
  
import type { CliStepMethod } from "embedding-sdk/cli/types/cli";  
  
const delay = (duration: number) =>  
 new Promise((resolve) => setTimeout(resolve, duration));  
  
const...

REQ-967: import ora from "ora";  
  
import { HARDCODED\_JWT\_SHARED\_SECRET } from "../constants/config";  
import { getEmbeddingFailedMessage } from "../constants/messages";  
import type { CliStepMethod } from...

REQ-968: import { input, select } from "@inquirer/prompts";  
import chalk from "chalk";  
import toggle from "inquirer-toggle";  
import open from "open";  
import ora from "ora";  
  
import type { CliStepMethod } from...

REQ-969: import ora from "ora";  
  
import { SITE\_NAME } from "../constants/config";  
import { EMBEDDING\_DEMO\_SETUP\_TOKEN } from "../constants/env";  
import { INSTANCE\_CONFIGURED\_MESSAGE } from...

REQ-970: import { SANDBOXED\_GROUP\_NAMES } from "../constants/config";  
import { getNoTenantMessage } from "../constants/messages";  
import type { CliStepMethod } from "../types/cli";  
import { createCollection }...

REQ-971: import { SDK\_DOCS\_LINK } from "embedding-sdk/cli/constants/config";  
  
import { SHOW\_ON\_STARTUP\_MESSAGE } from "../constants/messages";  
import type { CliStepMethod } from "../types/cli";  
import {...

REQ-972: import { select } from "@inquirer/prompts";  
import { green } from "chalk";  
  
import {  
 SDK\_LEARN\_MORE\_MESSAGE,  
 getMetabaseInstanceSetupCompleteMessage,  
} from "../constants/messages";  
import type {...

REQ-973: import { exec as execCallback } from "child\_process";  
  
import chalk from "chalk";  
import ora from "ora";  
import { promisify } from "util";  
  
import {  
 CONTAINER\_NAME,  
 DEFAULT\_PORT,  
 IMAGE\_NAME,  
}...

REQ-974: import type { CLI\_STEPS } from "embedding-sdk/cli/run";  
import type { Settings, Table } from "metabase-types/api";  
  
import type { DashboardInfo } from "../types/dashboard";  
  
export type CliState =...

REQ-975: import type { DashboardId } from "metabase-types/api";  
  
export type DashboardInfo = { id: DashboardId; name: string };

REQ-976: import { propagateErrorResponse } from "./propagate-error-response";  
  
interface Options {  
 name: string;  
 engine: string;  
 connection: Record<string, string | boolean | number>;  
  
 cookie:...

REQ-977: import fs from "fs/promises";  
  
const GITIGNORE\_PATH = ".gitignore";  
  
/\*\*  
 \* Adds the credential file to .gitignore if exists.  
 \*/  
export async function addFileToGitIgnore(fileName: string) {  
 try {  
...

REQ-978: import fs from "fs/promises";  
  
import { input, number, password, select } from "@inquirer/prompts";  
import { EventEmitter } from "events";  
import fileSelector from "inquirer-file-selector";  
import...

REQ-979: import fs from "fs";  
import path from "path";  
  
import { getProjectDependenciesFromPackageJson } from "../utils/get-package-version";  
  
/\*\*  
 \* Checks if the current project is a TypeScript project.  
 \*  
...

REQ-980: import { propagateErrorResponse } from "embedding-sdk/cli/utils/propagate-error-response";  
  
interface Options {  
 name: string;  
  
 instanceUrl: string;  
 cookie: string;  
}  
  
export async function...

REQ-981: import type { Table } from "metabase-types/api";  
  
import { propagateErrorResponse } from "./propagate-error-response";  
  
interface Options {  
 table: Table;  
 databaseId: number;  
 collectionId:...

REQ-982: import type { Settings } from "metabase-types/api";  
  
interface Options {  
 instanceUrl: string;  
}  
  
export async function fetchInstanceSettings(  
 options: Options,  
): Promise<Settings | null> {  
 ...

REQ-983: export function generateRandomDemoPassword(): string {  
 const chars = "abcdefghijklmnopqrstuvwxyz";  
 const upperCaseChars = chars.toUpperCase();  
 const numbers = "0123456789";  
  
 const allChars =...

REQ-984: const PORT\_REGEX = /(?:0\.0\.0\.0:|:::)(\d+)->3000\/tcp/g;  
  
export function getCurrentDockerPort(ports: string): number | null {  
 if (!ports || ports.length === 0) {  
 return null;  
 }  
  
 let...

REQ-985: import path from "path";  
  
import { GENERATED\_COMPONENTS\_DEFAULT\_PATH } from "../constants/config";  
  
export const getExampleComponentImportPath = (  
 reactComponentDir =...

REQ-986: import { getExampleComponentImportPath } from "./get-example-component-import-path";  
  
describe("CLI > getExampleImportPath", () => {  
 it("should return a valid example import path", () => {  
 ...

REQ-987: import { exec as execCallback } from "child\_process";  
  
import { promisify } from "util";  
  
import { safeJsonParse } from "metabase/lib/json-parse";  
  
import { CONTAINER\_NAME } from...

REQ-988: import { green } from "chalk";  
  
import { NEXTJS\_DEMO\_ROUTE\_NAME } from "../constants/config";  
import {  
 LINK\_TO\_NEXT\_JS\_GUIDE,  
 LINK\_TO\_NEXT\_JS\_SAMPLE,  
} from "../constants/messages";  
import {...

REQ-989: import fs from "fs/promises";  
import path from "path";  
  
export const hasPackageJson = async () => {  
 try {  
 await fs.access("package.json");  
 return true;  
 } catch (error) {  
 return...

REQ-990: import {  
 DataPermission,  
 DataPermissionValue,  
} from "metabase/admin/permissions/types";  
import type {  
 DatabasePermissions,  
 GroupsPermissions,  
 Table,  
} from "metabase-types/api";  
  
type...

REQ-991: interface Options {  
 groupIds: number[];  
 collectionIds: number[];  
}  
  
const ALL\_USERS\_GROUP\_ID = 1;  
  
export function getSandboxedCollectionPermissions(options: Options) {  
 const { groupIds,...

REQ-992: import type {  
 FieldReference,  
 GroupTableAccessPolicy,  
 Table,  
} from "metabase-types/api";  
  
type Options = {  
 groupIds: number[];  
 chosenTables: Table[];  
 tenancyColumnNames: Record<string,...

REQ-993: import { exec as execCallback } from "child\_process";  
import path from "path";  
  
import { detect } from "detect-package-manager";  
import ora from "ora";  
import { match } from "ts-pattern";  
import {...

REQ-994: import { type Server, createServer } from "net";  
  
/\*\*  
 \* Check if a port is taken.  
 \* Creates a TCP server on the given port and waits for it to be closed.  
 \*/  
export const checkIsPortTaken = (port:...

REQ-995: import fs from "fs";  
import path from "path";  
  
import { glob } from "glob";  
  
import { NEXTJS\_DEMO\_ROUTE\_NAME } from "../constants/config";  
import {  
 getNextJsAnalyticsPageSnippet,  
 ...

REQ-996: import { createMockField, createMockTable } from "metabase-types/api/mocks";  
  
import { getPermissionsForGroups } from "./get-permission-groups";  
import { getSandboxedCollectionPermissions } from...

REQ-997: import chalk from "chalk";  
  
const MAX\_WIDTH = 80;  
  
export const OUTPUT\_STYLES = {  
 title: chalk.bold.bgHex("#509EE3").white,  
 version: chalk.hex("#509EE3"),  
 link: chalk.underline.blueBright,  
 ...

REQ-998: import type { CliError } from "../types/cli";  
  
// Propagate the error from the API to the CLI.  
export const propagateErrorResponse = async (res: Response) => {  
 if (res.ok) {  
 return;  
 }  
  
 let...

REQ-999: const waitFor = (delayMs: number) =>  
 new Promise((resolve) => setTimeout(resolve, delayMs));  
  
export async function retry<T>(  
 task: () => Promise<T>,  
 options?: {  
 retries?: number;  
 ...

REQ-1000: import type { Dataset, FieldReference, Table } from "metabase-types/api";  
  
import { propagateErrorResponse } from "./propagate-error-response";  
  
interface Options {  
 table: Table;  
 columnName:...

REQ-1001: import { select } from "@inquirer/prompts";  
  
import { CONTINUE\_SETUP\_ON\_WARNING\_MESSAGE } from "../constants/messages";  
  
import { printWarning } from "./print";  
  
/\*\*  
 \* @returns {boolean} whether the...

REQ-1002: import { GENERATED\_COMPONENTS\_DEFAULT\_PATH } from "../constants/config";  
  
/\*\*  
 \* Where should we save the generated components by default?  
 \*/  
export const getGeneratedComponentsDefaultPath = ({  
 ...

REQ-1003: import type {  
 MetabaseDashboard,  
 SdkDashboardId,  
} from "embedding-sdk/types/dashboard";  
import { uuid } from "metabase/lib/uuid";  
  
import { propagateErrorResponse } from...

REQ-1004: export \* from "./PublicComponentWrapper";  
export \* from "./withPublicComponentWrapper";  
export \* from "./SdkError";  
export \* from "./SdkLoader";

REQ-1005: export { SdkAdHocQuestion } from "./SdkAdHocQuestion";

REQ-1006: export \* from "./BackButton";

REQ-1007: export \* from "./BreakoutDropdown";

REQ-1008: export \* from "./BreakoutPicker";

REQ-1009: export \* from "./Breakout";  
export \* from "./BreakoutDropdown";  
export \* from "./BreakoutPicker";

REQ-1010: import { isNotNull } from "metabase/lib/types";  
import {  
 type ListItem as BreakoutListItem,  
 getBreakoutListItem,  
} from...

REQ-1011: export \* from "./ChartTypeDropdown";

REQ-1012: export \* from "./DownloadWidget";  
export \* from "./DownloadWidgetDropdown";

REQ-1013: export \* from "./EditorButton";

REQ-1014: export \* from "./FilterDropdown";

REQ-1015: export \* from "./FilterPicker";

REQ-1016: import { useCallback, useMemo } from "react";  
  
import { useSdkQuestionContext } from "embedding-sdk/components/private/SdkQuestion/context";  
import type { FilterItem } from...

REQ-1017: import type { UpdateQueryHookProps } from "metabase/query\_builder/hooks";  
import \* as Lib from "metabase-lib";  
  
export const useFilterHandlers = ({  
 query,  
 stageIndex = -1,  
 onQueryChange,  
}:...

REQ-1018: export \* from "./FilterDropdown";  
export \* from "./Filter";  
export \* from "./FilterPicker";

REQ-1019: export \* from "./QuestionSettingsDropdown";

REQ-1020: export \* from "./QuestionSettings";  
export \* from "./QuestionSettingsDropdown";

REQ-1021: export \* from "./ResetButton";

REQ-1022: export \* from "./SummarizeDropdown";

REQ-1023: export \* from "./SummarizePicker";

REQ-1024: export \* from "./Summarize";  
export \* from "./SummarizeDropdown";  
export \* from "./SummarizePicker";

REQ-1025: import { useCallback, useMemo } from "react";  
  
import {  
 type AggregationItem,  
 getAggregationItems,  
} from "metabase/query\_builder/utils/get-aggregation-items";  
import \* as Lib from...

REQ-1026: export \* from "./VisualizationButton";

REQ-1027: export \* from "./BackButton";  
export \* from "./Breakout";  
export \* from "./ChartTypeSelector";  
export \* from "./ChartTypeSelectorList";  
export \* from "./DownloadWidget";  
export \* from...

REQ-1028: export \* from "./AddBadgeListItem";

REQ-1029: export \* from "./BadgeListItem";

REQ-1030: export \* from "./BadgeList";

REQ-1031: export \* from "./ToolbarButton";

REQ-1032: export \* from "./SdkQuestionProvider";  
export \* from "./types";

REQ-1033: import type { PropsWithChildren } from "react";  
  
import type { LoadQuestionHookResult } from "embedding-sdk/hooks/private/use-load-question";  
import type { SdkCollectionId } from...

REQ-1034: import { useQuestionVisualizationState } from "metabase/query\_builder/components/chart-type-selector";  
  
import { useSdkQuestionContext } from "../context";  
  
export const useQuestionVisualization = ()...

REQ-1035: import { useMemo } from "react";  
  
import { transformSdkQuestion } from "embedding-sdk/lib/transform-question";  
import { isQuestionDirty } from "metabase/query\_builder/utils/question";  
import {...

REQ-1036: import { useMemo } from "react";  
  
import { getSensibleVisualizations } from "metabase/query\_builder/components/chart-type-selector";  
  
import { useSdkQuestionContext } from "../context";  
  
export const...

REQ-1037: export \* from "./SdkQuestionDefaultView";

REQ-1038: export \* from "./MultiStepPopover";

REQ-1039: export \* from "./CollectionBrowser";

REQ-1040: export { CreateDashboardModal } from "./CreateDashboardModal";  
export type { CreateDashboardModalProps } from "./CreateDashboardModal";

REQ-1041: export \* from "./CreateQuestion";

REQ-1042: export \* from "./InteractiveQuestion";

REQ-1043: export { MetabotQuestion } from "./MetabotQuestion";

REQ-1044: export {  
 type BackButtonProps as InteractiveQuestionBackButtonProps,  
 type InteractiveQuestionBreakoutDropdownProps,  
 type ChartTypeDropdownProps as InteractiveQuestionChartTypeDropdownProps,  
 ...

REQ-1045: import type { SdkQuestionId } from "embedding-sdk/types/question";  
  
export type SdkQuestionIdProps = {  
 /\*\*  
 \* The ID of the question.  
 \* <br/>  
 \* This is either:  
 \* <br/>  
 \* - The...

REQ-1046: export \* from "./StaticQuestion";

REQ-1047: import type { QueryClickActionsMode } from "metabase/visualizations/types";  
  
export const StaticQuestionSdkMode: QueryClickActionsMode = {  
 name: "static-question-sdk",  
 hasDrills: false,  
 ...

REQ-1048: export {  
 EditableDashboard,  
 type EditableDashboardProps,  
} from "./EditableDashboard";

REQ-1049: export {  
 InteractiveDashboard,  
 type InteractiveDashboardProps,  
} from "./InteractiveDashboard";

REQ-1050: export { StaticDashboard, type StaticDashboardProps } from "./StaticDashboard";

REQ-1051: export \* from "./EditableDashboard";  
export \* from "./InteractiveDashboard";  
export \* from "./StaticDashboard";

REQ-1052: export { CollectionBrowser } from "./CollectionBrowser";  
export { CreateDashboardModal } from "./CreateDashboardModal";  
export { CreateQuestion } from "./CreateQuestion";  
export {  
 StaticDashboard,  
...

REQ-1053: export const DEFAULT\_FONT = "Lato";  
  
export const getEmbeddingSdkVersion = (): string | "unknown" =>  
 (process.env.EMBEDDING\_SDK\_VERSION as string) ?? "unknown";

REQ-1054: eslint-disable-next-line import/no-commonjs

REQ-1055: Docs for development on the sdk

REQ-1056: export class MetabaseError<C extends string, P> extends Error {  
 public readonly code: C;  
 public readonly params?: P;  
  
 constructor(code: C, message: string, params?: P) {  
 super(message);  
 ...

REQ-1057: import { MetabaseError } from "./base";  
  
export function USER\_FETCH\_FAILED() {  
 return new MetabaseError(  
 "USER\_FETCH\_FAILED",  
 "Failed to fetch the user, the session might be invalid.",  
 ...

REQ-1058: export \* from "./base";  
export \* from "./jwt";  
export \* from "./version";  
export \* from "./saml";  
export \* from "./generic";

REQ-1059: import { MetabaseError } from "./base";  
  
export function INVALID\_SESSION\_OBJECT(params: {  
 expected?: string;  
 actual?: string;  
}) {  
 return new MetabaseError(  
 "INVALID\_SESSION\_OBJECT",  
 ...

REQ-1060: import { MetabaseError } from "./base";  
  
export function SAML\_POPUP\_BLOCKED() {  
 return new MetabaseError(  
 "SAML\_POPUP\_BLOCKED",  
 "Popup blocked. Please allow popups for this site.",  
 ...

REQ-1061: import { MetabaseError } from "./base";  
  
export function SDK\_VERSION\_INCOMPATIBLE(params: {  
 expected?: string;  
 actual?: string;  
}) {  
 return new MetabaseError(  
 "SDK\_VERSION\_INCOMPATIBLE",  
 ...

REQ-1062: export \* from "./private";

REQ-1063: export \* from "./use-init-data";

REQ-1064: export \* from "./use-init-data";

REQ-1065: import { useEffect, useRef } from "react";  
import { useMount } from "react-use";  
import \_ from "underscore";  
  
import { getEmbeddingSdkVersion } from "embedding-sdk/config";  
import { useLazySelector }...

REQ-1066: import { useReducer, useRef, useState } from "react";  
import { useAsyncFn, useUnmount } from "react-use";  
  
import {  
 loadQuestionSdk,  
 runQuestionOnNavigateSdk,  
 runQuestionQuerySdk,  
 ...

REQ-1067: import { pick } from "underscore";  
  
import type { SdkDashboardId } from "embedding-sdk/types/dashboard";  
import type { CommonStylingProps } from "embedding-sdk/types/props";  
import {...

REQ-1068: import { useElementSize } from "@mantine/hooks";  
  
import { getDefaultVizHeight } from "embedding-sdk/lib/default-height";  
import type { VisualizationDisplay } from "metabase-types/api";  
  
export const...

REQ-1069: import { useEffect, useMemo, useRef } from "react";  
  
import { printUsageProblemToConsole } from "embedding-sdk/lib/print-usage-problem";  
import { getSdkUsageProblem } from...

REQ-1070: import { isValidId } from "embedding-sdk/lib/is-valid-collection-id";  
import { getCollectionIdSlugFromReference } from "embedding-sdk/store/collections";  
import type { SdkCollectionId } from...

REQ-1071: import { EMBEDDING\_SDK\_CONFIG } from "metabase/embedding-sdk/config";  
import { defineGlobalDependencies } from "metabase/embedding-sdk/lib/define-global-dependencies";  
  
// Enable SDK mode as we are...

REQ-1072: const RESTRICTED\_CONSOLE\_PATTERNS = [  
 /UNSAFE\_component.\*Visualization/,  
 /UNSAFE\_component.\*DashboardGrid/,  
 /Warning: React does not recognize the `.\*?` prop on a DOM element/,  
 /Warning:...

REQ-1073: import { ensureMetabaseProviderPropsStore } from "embedding-sdk/sdk-shared/lib/ensure-metabase-provider-props-store";  
  
afterEach(() => {  
 ensureMetabaseProviderPropsStore().cleanup();  
});

REQ-1074: import { EMBEDDING\_SDK\_CONFIG } from "metabase/embedding-sdk/config";  
  
// eslint-disable-next-line no-undef  
process.env.IS\_EMBEDDING\_SDK = "true";  
  
EMBEDDING\_SDK\_CONFIG.isEmbeddingSdk = true;

REQ-1075: import { getDefaultSize } from "metabase/visualizations/shared/utils/sizes";  
import type { VisualizationDisplay } from "metabase-types/api";  
  
/\*\*  
 \* How many pixels are in each cell?  
 \*  
 \* The card...

REQ-1076: export const getIsLocalhost = () => {  
 const { hostname } = window.location;  
  
 return hostname === "localhost" || hostname === "127.0.0.1";  
};

REQ-1077: import type { CollectionId } from "metabase-types/api";  
  
export const isValidId = (  
 collectionId: unknown,  
): collectionId is CollectionId => {  
 return (  
 !!collectionId &&  
 (typeof...

REQ-1078: import type { Deferred } from "metabase/lib/promise";  
import { CardApi } from "metabase/services";  
import type { Card, Dataset, ParameterQueryObject } from "metabase-types/api";  
  
interface Options {  
...

REQ-1079: eslint-disable no-color-literals  
Note: these functions need to return an array of two elements, when styling  
console.logs, the style needs to passed as second argument  
To use them, do...

REQ-1080: import { merge } from "icepick";  
  
import type {  
 MetabaseDashboardPluginsConfig,  
 MetabasePluginsConfig,  
} from "embedding-sdk/types/plugins";  
  
const DEFAULT\_DASHCARD\_MENU\_ITEMS:...

REQ-1081: import React from "react";  
import { useSyncExternalStore } from "use-sync-external-store/shim";  
  
// Monkey-patches useSyncExternalStore if we are in React 17,  
// where useSyncExternalStore is not...

REQ-1082: eslint-disable no-color-literals

REQ-1083: export { updateQuestionSdk } from "./update-question";  
export { loadQuestionSdk } from "./load-question";  
export { runQuestionOnNavigateSdk } from "./run-question-on-navigate";  
export {...

REQ-1084: import \_ from "underscore";  
  
import type { LoadSdkQuestionParams } from "embedding-sdk/types/question";  
import { resolveCards } from "metabase/query\_builder/actions";  
import {...

REQ-1085: import { runQuestionQuerySdk } from "embedding-sdk/lib/sdk-question/run-question-query";  
import type {  
 NavigateToNewCardParams,  
 SdkQuestionState,  
} from "embedding-sdk/types/question";  
import {...

REQ-1086: import type { SdkQuestionState } from "embedding-sdk/types/question";  
import type { Deferred } from "metabase/lib/promise";  
import { runQuestionQuery } from "metabase/services";  
import {...

REQ-1087: import \_ from "underscore";  
  
import type { SdkQuestionState } from "embedding-sdk/types/question";  
import type { Deferred } from "metabase/lib/promise";  
import { computeQuestionPivotTable } from...

REQ-1088: \*

REQ-1089: type ColorTuple = [  
 string,  
 string,  
 string,  
 string,  
 string,  
 string,  
 string,  
 string,  
 string,  
 string,  
];  
  
export const colorTuple = (value: string): ColorTuple =>  
 ...

REQ-1090: import { getEmbeddingColorPalette } from "metabase/embedding-sdk/theme/embedding-color-palette";  
  
describe("Embedding Color Palette", () => {  
 it("transforms chart color overrides into accent...

REQ-1091: import { merge } from "icepick";  
import \_ from "underscore";  
  
import { DEFAULT\_FONT } from "embedding-sdk/config";  
import type {  
 MetabaseColor,  
 MetabaseComponentTheme,  
 MetabaseTheme,  
} from...

REQ-1092: import {  
 DEFAULT\_EMBEDDED\_COMPONENT\_THEME,  
 getEmbeddingComponentOverrides,  
} from "metabase/embedding-sdk/theme";  
  
import { getEmbeddingThemeOverride } from...

REQ-1093: export { getEmbeddingThemeOverride } from "./get-embedding-theme";

REQ-1094: import type { MetabaseQuestion } from "metabase/embedding-sdk/types/question";  
import type Question from "metabase-lib/v1/Question";  
  
export function transformSdkQuestion(question: Question):...

REQ-1095: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import { transformSdkQuestion } from "embedding-sdk/lib/transform-question";  
import Question from "metabase-lib/v1/Question";  
import {...

REQ-1096: import { match } from "ts-pattern";  
  
import type { MetabaseAuthConfig } from "embedding-sdk/types";  
import type {  
 SdkUsageProblem,  
 SdkUsageProblemKey,  
} from...

REQ-1097: import { versionToNumericComponents } from "metabase/lib/utils";  
  
// They are mostly used for local development  
export const isInvalidMetabaseVersion = (mbVersion: string) => {  
 return (  
 ...

REQ-1098: import {  
 isInvalidMetabaseVersion,  
 isSdkVersionCompatibleWithMetabaseVersion,  
} from "./version-utils";  
  
const expectCompatibility = ({  
 mbVersion,  
 sdkVersion,  
 expected,  
}: {  
 mbVersion:...

REQ-1099: export \* from "./CollectionBrowser";

REQ-1100: export \* from "./CreateDashboardModal";

REQ-1101: export \* from "./CreateQuestion";

REQ-1102: export \* from "./InteractiveQuestion";

REQ-1103: eslint-disable-next-line no-literal-metabase-strings -- Export

REQ-1104: export \* from "./MetabotQuestion";

REQ-1105: export \* from "./StaticQuestion";

REQ-1106: export \* from "./EditableDashboard";

REQ-1107: export \* from "./InteractiveDashboard";

REQ-1108: export \* from "./StaticDashboard";

REQ-1109: export \* from "./SdkDebugInfo";

REQ-1110: export const SDK\_BUNDLE\_SCRIPT\_DATA\_ATTRIBUTE\_DASH\_CASED =  
 "embedding-sdk-bundle";  
export const SDK\_BUNDLE\_SCRIPT\_DATA\_ATTRIBUTE\_PASCAL\_CASED =  
 "embeddingSdkBundle";  
  
export const...

REQ-1111: import { useEffect, useMemo } from "react";  
  
import { useMetabaseProviderPropsStore } from "embedding-sdk/sdk-shared/hooks/use-metabase-provider-props-store";  
import {  
 type...

REQ-1112: import { useCallback, useSyncExternalStore } from "react";  
  
import { useMetabaseProviderPropsStore } from "embedding-sdk/sdk-shared/hooks/use-metabase-provider-props-store";  
import type {...

REQ-1113: import { useEffect } from "react";  
  
// eslint-disable-next-line no-external-references-for-sdk-package-code  
import { SDK\_BUNDLE\_FULL\_PATH } from...

REQ-1114: import { useLazySelector } from "embedding-sdk/sdk-package/hooks/private/use-lazy-selector";  
import { getWindow } from "embedding-sdk/sdk-shared/lib/get-window";  
  
/\*\*  
 \* Returns application name.  
 \*...

REQ-1115: import { useMemo } from "react";  
  
import { useLazySelector } from "embedding-sdk/sdk-package/hooks/private/use-lazy-selector";  
import { getWindow } from...

REQ-1116: export { useCreateDashboardApi } from "./use-create-dashboard-api";

REQ-1117: import { useCallback, useMemo } from "react";  
  
import { useLazySelector } from "embedding-sdk/sdk-package/hooks/private/use-lazy-selector";  
import { useMetabaseProviderPropsStore } from...

REQ-1118: import { useLazySelector } from "embedding-sdk/sdk-package/hooks/private/use-lazy-selector";  
import { getWindow } from "embedding-sdk/sdk-shared/lib/get-window";  
import type { MetabaseUser } from...

REQ-1119: import { useLazySelector } from "embedding-sdk/sdk-package/hooks/private/use-lazy-selector";  
import { getWindow } from "embedding-sdk/sdk-shared/lib/get-window";  
  
/\*\*  
 \* Returns the authentication...

REQ-1120: import { SDK\_BUNDLE\_SCRIPT\_DATA\_ATTRIBUTE\_DASH\_CASED } from "embedding-sdk/sdk-package/config";  
  
export function getSdkBundleScriptElement(): HTMLScriptElement | null {  
 return...

REQ-1121: import type { MetabaseAuthConfig } from "embedding-sdk/types";  
  
/\*\*  
 \* Defines a Metabase auth config.  
 \*  
 \* @function  
 \* @category MetabaseProvider  
 \*/  
export const defineMetabaseAuthConfig = (  
 ...

REQ-1122: eslint-disable-next-line no-external-references-for-sdk-package-code

REQ-1123: import { useEffect, useRef, useSyncExternalStore } from "react";  
  
import { ensureMetabaseProviderPropsStore } from "../lib/ensure-metabase-provider-props-store";  
  
export function...

REQ-1124: import { useMetabaseProviderPropsStore } from "embedding-sdk/sdk-shared/hooks/use-metabase-provider-props-store";  
import {  
 SdkLoadingError,  
 SdkLoadingState,  
} from...

REQ-1125: import { useCallback } from "react";  
  
import { useMetabaseProviderPropsStore } from "embedding-sdk/sdk-shared/hooks/use-metabase-provider-props-store";  
  
export const useSingleInstanceIdsData = () =>...

REQ-1126: import type { InternalMetabaseProviderProps } from "embedding-sdk/components/public/MetabaseProvider";  
import {  
 type SdkLoadingError,  
 SdkLoadingState,  
} from...

REQ-1127: export function getWindow() {  
 return typeof window !== "undefined" ? window : null;  
}

REQ-1128: export enum SdkLoadingError {  
 NotStartedLoading = "NotStartedLoading",  
 Error = "Error",  
}  
  
export enum SdkLoadingState {  
 Initial = 0,  
 Loading = 1,  
 Loaded = 2,  
 Initialized = 3,  
}

REQ-1129: import {  
 connectToInstanceAuthSso,  
 jwtDefaultRefreshTokenFunction,  
 openSamlLoginPopup,  
 samlTokenStorage,  
 validateSessionToken,  
} from "embedding/auth-common";  
import {...

REQ-1130: export \* from "./auth";  
export \* from "../../../embedding/auth-common/saml-token-storage";

REQ-1131: import { createSelector } from "@reduxjs/toolkit";  
import { P, match } from "ts-pattern";  
  
import { getUserPersonalCollectionId } from "metabase/selectors/user";  
import type { CollectionId,...

REQ-1132: eslint-disable no-restricted-imports

REQ-1133: import { createAction, createReducer } from "@reduxjs/toolkit";  
  
import { samlTokenStorage } from "embedding/auth-common";  
import type { SdkState, SdkStoreState } from...

REQ-1134: import type { SdkStoreState } from "embedding-sdk/store/types";  
import { getSetting } from "metabase/selectors/settings";  
import type { State } from "metabase-types/store";  
  
export const...

REQ-1135: import type {  
 Action,  
 AnyAction,  
 SerializedError,  
 Store,  
 ThunkDispatch,  
} from "@reduxjs/toolkit";  
import type { JSX } from "react";  
  
import type { MetabaseAuthConfig } from...

REQ-1136: import { useContext } from "react";  
import type { TypedUseSelectorHook } from "react-redux";  
import { createSelectorHook } from "react-redux";  
  
import type { SdkStoreState } from...

REQ-1137: import { renderHook } from "\_\_support\_\_/ui";  
  
import {  
 USE\_OUTSIDE\_OF\_CONTEXT\_MESSAGE,  
 useSdkSelector,  
} from "./use-sdk-selector";  
  
describe("useSdkSelector", () => {  
 it("should throw an error...

REQ-1138: {  
 "rules": {  
 "no-color-literals": "off"  
 }  
}

REQ-1139: import { EMBEDDING\_SDK\_CONFIG } from "metabase/embedding-sdk/config";  
  
describe("SDK environment config", () => {  
 beforeEach(() => {  
 EMBEDDING\_SDK\_CONFIG.isEmbeddingSdk = false;  
 });  
  
 ...

REQ-1140: import type {  
 MetabaseAuthConfig,  
 MetabaseAuthConfigWithApiKey,  
} from "embedding-sdk/types";  
  
import { MOCK\_INSTANCE\_URL } from "./sso";  
  
export const createMockSdkConfig = (  
 opts:...

REQ-1141: import fetchMock from "fetch-mock";  
  
// ===== MOCK CONSTANTS =====  
  
export const MOCK\_INSTANCE\_URL = "http://localhost";  
export const MOCK\_JWT\_PROVIDER\_URI = "http://test\_uri/sso/metabase";  
export...

REQ-1142: import type {  
 EmbeddingSessionTokenState,  
 SdkState,  
} from "embedding-sdk/store/types";  
import type { LoginStatus } from "embedding-sdk/types/user";  
  
export const createMockTokenState = ({  
 ...

REQ-1143: import {  
 setupCurrentUserEndpoint,  
 setupPropertiesEndpoints,  
 setupSettingsEndpoints,  
} from "\_\_support\_\_/server-mocks";  
import { mockSettings } from "\_\_support\_\_/settings";  
import type {...

REQ-1144: const QUESTION\_ENTITY\_ID = "\_GiVL6zYmsnBb1oqLCp4u";  
const DASHBOARD\_ENTITY\_ID = "xBLdW9FsgRuB2HGhWiBa\_";  
const COLLECTION\_ENTITY\_ID = "HyB3nRtqb7pBPhFG26evI";  
  
export const questionIds =...

REQ-1145: import {  
 type MetabaseTheme,  
 defineMetabaseTheme,  
} from "metabase/embedding-sdk/theme";  
  
export const darkColors = {  
 primary: "#DF75E9",  
 filter: "#7ABBF9",  
 lighterGrey: "#E3E7E4",  
 ...

REQ-1146: import type { MetabaseFetchRequestTokenFn } from "embedding-sdk/types/refresh-token";  
  
/\*\*  
 \* @inline  
 \*/  
type BaseMetabaseAuthConfig = {  
 metabaseInstanceUrl: string;  
};  
  
/\*\*  
 \* @category...

REQ-1147: import type { SdkUserId } from "embedding-sdk/types/user";  
  
import type { SdkEntityId } from "./entity-id";  
  
// "CollectionId" from core app also includes "root" | "users" and "trash", we don't want...

REQ-1148: import type {  
 MetabaseCollection,  
 SdkCollectionId,  
} from "embedding-sdk/types/collection";  
import type { CreateDashboardProperties } from...

REQ-1149: eslint-disable-next-line @typescript-eslint/ban-types -- allows for other string literal types in other ID types

REQ-1150: import type { MetabaseDashboard } from "embedding-sdk/types/dashboard";  
  
export type SdkDashboardLoadEvent = (  
 dashboard: MetabaseDashboard | null,  
) => void;  
  
export type SdkEventHandlersConfig =...

REQ-1151: eslint-disable @typescript-eslint/consistent-type-imports

REQ-1152: export type \* from "./auth-config";  
export type \* from "./collection";  
export type \* from "./dashboard";  
export type \* from "./entity-id";  
export type \* from "./events";  
export type \* from...

REQ-1153: import type { JSX, ReactNode } from "react";  
  
import type { MetabaseTheme } from "metabase/embedding-sdk/theme";  
  
import type { MetabaseAuthConfig } from "./auth-config";  
import type {...

REQ-1154: import type { DashboardCardMenu } from "metabase/dashboard/components/DashCard/DashCardMenu/dashcard-menu";  
  
export type MetabaseClickAction = {  
 name: string;  
} & Record<string, any>;  
  
export type...

REQ-1155: import type { CSSProperties } from "react";  
  
/\*\*  
 \* @inline  
 \*/  
export type CommonStylingProps = {  
 /\*\*  
 \* A custom class name to be added to the root element.  
 \*/  
 className?: string;  
  
 /\*\*  
 ...

REQ-1156: import type { ReactNode } from "react";  
  
import type { Deferred } from "metabase/lib/promise";  
import type { QueryParams } from "metabase/query\_builder/actions";  
import type { ObjectId } from...

REQ-1157: export type MetabaseEmbeddingSessionToken = {  
 id: string;  
 exp: number;  
};  
  
/\*\*  
 \* @inline  
 \*/  
export type UserBackendJwtResponse = {  
 jwt: string;  
};  
  
export type MetabaseFetchRequestTokenFn =...

REQ-1158: import type { JSX, ReactNode } from "react";  
  
export type { ButtonProps } from "metabase/ui";  
export type {  
 ChartColor,  
 MetabaseTheme,  
 MetabaseColors,  
 MetabaseComponentTheme,  
} from...

REQ-1159: import type { USAGE\_PROBLEM\_MESSAGES } from "embedding-sdk/lib/usage-problem";  
  
export interface SdkUsageProblem {  
 type: SdkUsageProblemKey;  
 severity: "warning" | "error";  
 title: string;  
 ...

REQ-1160: export type SdkUserId = number;  
  
/\*\*  
 \* The User entity  
 \*/  
export type MetabaseUser = {  
 id: SdkUserId;  
 first\_name: string | null;  
 last\_name: string | null;  
 common\_name: string;  
 email:...

REQ-1161: export \* from "./ImpersonationModal";

REQ-1162: export \* from "./ImpersonationWarning";

REQ-1163: import \_ from "underscore";  
  
import type { EntityId } from "metabase/admin/permissions/types";  
import {  
 DataPermission,  
 DataPermissionValue,  
} from "metabase/admin/permissions/types";  
import {  
 ...

REQ-1164: import {  
 DataPermission,  
 DataPermissionValue,  
} from "metabase/admin/permissions/types";  
import Database from "metabase-lib/v1/metadata/Database";  
import Schema from...

REQ-1165: import { push } from "react-router-redux";  
import { t } from "ttag";  
  
import { DataPermissionValue } from "metabase/admin/permissions/types";  
import {  
 getDatabaseFocusPermissionsUrl,  
 ...

REQ-1166: import type { PayloadAction } from "@reduxjs/toolkit";  
import { createAction, createSlice } from "@reduxjs/toolkit";  
import { push } from "react-router-redux";  
  
import {  
 LOAD\_DATA\_PERMISSIONS,  
 ...

REQ-1167: import type { DatabaseId, GroupId } from "metabase-types/api";  
  
import type { AdvancedPermissionsStoreState } from "./types";  
  
export const getImpersonation =  
 (databaseId: DatabaseId, groupId:...

REQ-1168: import { GET } from "metabase/lib/api";  
  
export const ImpersonationApi = {  
 get: GET("/api/ee/advanced-permissions/impersonation"),  
};

REQ-1169: import type { PartialBy } from "metabase/common/types";  
import type { EnterpriseSharedState } from "metabase-enterprise/shared/reducer";  
import type { EnterpriseState } from...

REQ-1170: import type { EntityId } from "metabase/admin/permissions/types";  
import {  
 getDatabaseFocusPermissionsUrl,  
 getGroupFocusPermissionsUrl,  
} from "metabase/admin/permissions/utils/urls";  
import type...

REQ-1171: import { setSidebar } from "metabase/dashboard/actions";  
import { SIDEBAR\_NAME } from "metabase/dashboard/constants";  
import type { DashCardId } from "metabase-types/api";  
import type { Dispatch }...

REQ-1172: export { AIQuestionAnalysisSidebar } from "./AIQuestionAnalysisSidebar";

REQ-1173: import { isNotNull } from "metabase/lib/types";  
import type { CollectionId, Timeline, TimelineEvent } from "metabase-types/api";  
  
export const getTimelineEventsForAnalysis = (  
 ...

REQ-1174: import { useEffect, useRef } from "react";  
import { usePrevious } from "react-use";  
  
import {  
 getChartImagePngDataUri,  
 getChartSelector,  
} from "metabase/visualizations/lib/image-exports";  
import...

REQ-1175: import { t } from "ttag";  
  
import {  
 PLUGIN\_AI\_ENTITY\_ANALYSIS,  
 PLUGIN\_DASHCARD\_MENU,  
} from "metabase/plugins";  
import { hasPremiumFeature } from "metabase-enterprise/settings";  
import type...

REQ-1176: import type { VisualizationDisplay } from "metabase-types/api";  
  
const CHART\_ANALYSIS\_ENABLED = {  
 // enabled  
 area: true,  
 bar: true,  
 combo: true,  
 funnel: true,  
 gauge: true,  
 line: true,  
 ...

REQ-1177: export \* from "./FixSqlQueryButton";

REQ-1178: import { PLUGIN\_AI\_SQL\_FIXER } from "metabase/plugins";  
import { hasPremiumFeature } from "metabase-enterprise/settings";  
  
import { FixSqlQueryButton } from "./components/FixSqlQueryButton";  
  
if...

REQ-1179: export \* from "./GenerateSqlQueryButton";

REQ-1180: import { PLUGIN\_AI\_SQL\_GENERATION } from "metabase/plugins";  
import { hasPremiumFeature } from "metabase-enterprise/settings";  
  
import { GenerateSqlQueryButton } from...

REQ-1181: import { t } from "ttag";  
  
export function getPlaceholderText() {  
 return t`Write and select text to generate SQL with Metabot, or type SQL directly`;  
}

REQ-1182: import type {  
 AIEntityAnalysisResponse,  
 AIQuestionAnalysisParams,  
} from "metabase-types/api";  
  
import { EnterpriseApi } from "./api";  
  
const DEFAULT\_TIMEOUT = 30000;  
  
export const...

REQ-1183: import type {  
 GenerateSqlQueryRequest,  
 GenerateSqlQueryResponse,  
} from "metabase-types/api";  
  
import { EnterpriseApi } from "./api";  
  
export const aiSqlGenerationApi =...

REQ-1184: export { aiStreamingQuery, getInflightRequestsForUrl } from "./requests";  
export type { JSONValue } from "./types";

REQ-1185: import { match } from "ts-pattern";  
import \_ from "underscore";  
  
import type { MetabotHistory } from "metabase-types/api";  
  
import {  
 type KnownDataPart,  
 dataPartSchema,  
 finishPartSchema,  
 ...

REQ-1186: import { processChatResponse } from "./process-stream";  
import { createMockReadableStream } from "./test-utils";  
  
const getMockedCallbacks = () => ({  
 onTextPart: jest.fn(),  
 onDataPart:...

REQ-1187: import { nanoid } from "@reduxjs/toolkit";  
  
import api from "metabase/lib/api";  
  
import { type AIStreamingConfig, processChatResponse } from "./process-stream";  
import type { JSONValue } from...

REQ-1188: import fetchMock from "fetch-mock";  
  
import { aiStreamingQuery, getInflightRequestsForUrl } from "./requests";  
import { mockStreamedEndpoint } from "./test-utils";  
  
const ENDPOINT =...

REQ-1189: import \* as Yup from "yup";  
  
export const dataPartSchema = Yup.object({  
 type: Yup.string().required(),  
 version: Yup.number().required(),  
 value: Yup.mixed(),  
});  
  
export const knownDataPartTypes...

REQ-1190: import { defer } from "metabase/lib/promise";  
  
async function delay(timeout: number) {  
 return new Promise((resolve) => {  
 setTimeout(() => resolve(undefined), timeout);  
 });  
}  
  
export function...

REQ-1191: \*

REQ-1192: import { Api } from "metabase/api";  
  
import { ENTERPRISE\_TAG\_TYPES } from "./tags";  
  
export const EnterpriseApi = Api.enhanceEndpoints({  
 addTagTypes: ENTERPRISE\_TAG\_TYPES,  
});

REQ-1193: import type { AuditInfo } from "metabase-enterprise/audit\_app/types/state";  
  
import { EnterpriseApi } from "./api";  
  
export const auditInfoApi = EnterpriseApi.injectEndpoints({  
 endpoints: (builder)...

REQ-1194: import type { BillingInfo } from "metabase-types/api";  
  
import { EnterpriseApi } from "./api";  
  
export const billingInfoApi = EnterpriseApi.injectEndpoints({  
 endpoints: (builder) => ({  
 ...

REQ-1195: import { provideCollectionItemListTags } from "metabase/api/tags";  
import type {  
 ListStaleCollectionItemsRequest,  
 ListStaleCollectionItemsResponse,  
} from...

REQ-1196: import { invalidateTags, listTag } from "metabase/api/tags";  
import { contentTranslationEndpoints } from "metabase-enterprise/content\_translation/constants";  
import type { DictionaryResponse } from...

REQ-1197: import { invalidateTags, tag } from "metabase/api/tags";  
import type { DatabaseId } from "metabase-types/api";  
  
import { EnterpriseApi } from "./api";  
  
export const DatabaseReplicationApi =...

REQ-1198: import \_ from "underscore";  
  
import { idTag, invalidateTags, listTag } from "metabase/api/tags";  
import type {  
 CreateDestinationDatabaseRequest,  
 Database,  
 UpdateDatabaseRouterRequest,  
} from...

REQ-1199: import type { DatabaseId, GdrivePayload } from "metabase-types/api";  
  
import { EnterpriseApi } from "./api";  
  
export const gdriveApi = EnterpriseApi.injectEndpoints({  
 endpoints: (builder) => ({  
 ...

REQ-1200: export \* from "./ai-entity-analysis";  
export \* from "./ai-sql-generation";  
export \* from "./audit-info";  
export \* from "./api";  
export \* from "./billing-info";  
export \* from "./metabot";  
export \*...

REQ-1201: import type {  
 DeleteSuggestedMetabotPromptRequest,  
 MetabotApiEntity,  
 MetabotEntity,  
 MetabotId,  
 MetabotInfo,  
 PaginationRequest,  
 PaginationResponse,  
 SuggestedMetabotPromptsRequest,  
 ...

REQ-1202: import { invalidateTags, tag } from "metabase/api/tags";  
import type { EnterpriseSettings } from "metabase-types/api";  
  
import { EnterpriseApi } from "./api";  
  
type SAMLSettings = Pick<  
 ...

REQ-1203: import type {  
 MaskedScimApiKey,  
 UnmaskedScimApiKey,  
} from "metabase-enterprise/user\_provisioning/types";  
  
import { EnterpriseApi } from "./api";  
  
export const scimApi =...

REQ-1204: import { invalidateTags, tag } from "metabase/api/tags";  
import type { EmailSMTPOverrideSettings } from "metabase-types/api";  
  
import { EnterpriseApi } from "./api";  
  
export const smtpOverrideApi =...

REQ-1205: import type { TagDescription } from "@reduxjs/toolkit/query";  
  
export const ENTERPRISE\_TAG\_TYPES = [  
 "scim",  
 "metabot",  
 "metabot-entities-list",  
 "metabot-prompt-suggestions",  
 ...

REQ-1206: import { invalidateTags, provideTableListTags, tag } from "metabase/api/tags";  
import type {  
 DeleteUploadTableRequest,  
 UploadManagementResponse,  
} from "metabase-types/api";  
  
import {...

REQ-1207: import { GET, PUT } from "metabase/lib/api";  
  
export const ApplicationPermissionsApi = {  
 graph: GET("/api/ee/advanced-permissions/application/graph"),  
 updateGraph:...

REQ-1208: import { t } from "ttag";  
  
export const APPLICATION\_PERMISSIONS\_OPTIONS = {  
 yes: {  
 // eslint-disable-next-line ttag/no-module-declaration -- see metabase#55045  
 label: t`Yes`,  
 value:...

REQ-1209: import { t } from "ttag";  
  
import {  
 PLUGIN\_ADMIN\_ALLOWED\_PATH\_GETTERS,  
 PLUGIN\_APPLICATION\_PERMISSIONS,  
 PLUGIN\_REDUCERS,  
} from "metabase/plugins";  
import { hasPremiumFeature } from...

REQ-1210: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1211: import { assocIn } from "icepick";  
import { t } from "ttag";  
  
import { getErrorMessage } from "metabase/api/utils";  
import {  
 combineReducers,  
 createAction,  
 createThunkAction,  
 handleActions,  
}...

REQ-1212: import { createSelector } from "@reduxjs/toolkit";  
import { getIn } from "icepick";  
import { t } from "ttag";  
import \_ from "underscore";  
  
import { UNABLE\_TO\_CHANGE\_ADMIN\_PERMISSIONS } from...

REQ-1213: import type { GroupId } from "metabase-types/api";  
  
export type ApplicationPermissionKey =  
 | "subscription"  
 | "monitoring"  
 | "setting";  
export type ApplicationPermissionValue = "yes" |...

REQ-1214: import type { State } from "metabase-types/store";  
  
import type { ApplicationPermissions } from "./permissions";  
import type { UserWithApplicationPermissions } from "./user";  
  
export interface...

REQ-1215: export type { UserWithApplicationPermissions } from "metabase/plugins";

REQ-1216: import type { AdminPathKey } from "metabase-types/store";  
  
import type { UserWithApplicationPermissions } from "./types/user";  
  
const canAccessMonitoringItems = (user?:...

REQ-1217: export \* from "./AuditTableVisualization";

REQ-1218: export { UnsubscribeUserModal } from "./UnsubscribeUserModal";

REQ-1219: import { t } from "ttag";  
  
import { ForwardRefLink } from "metabase/common/components/Link";  
import {  
 PLUGIN\_ADMIN\_USER\_MENU\_ITEMS,  
 PLUGIN\_ADMIN\_USER\_MENU\_ROUTES,  
 PLUGIN\_AUDIT,  
} from...

REQ-1220: import { t } from "ttag";  
  
export const viewsAndAvgExecutionTimeByDay = () => ({  
 card: {  
 name: t`Query views and speed per day`,  
 display: "line",  
 dataset\_query: {  
 type:...

REQ-1221: import { push } from "react-router-redux";  
import \_ from "underscore";  
  
export const getColumnName = (column) => column.remapped\_to || column.name;  
  
export const getRowValuesByColumns = (row, cols)...

REQ-1222: import type { CardId, CollectionId, DashboardId } from "metabase-types/api";  
  
export interface AuditInfo {  
 dashboard\_overview: DashboardId;  
 question\_overview: CardId;  
 custom\_reports:...

REQ-1223: import type { Database } from "metabase-types/api";  
  
export const isAuditDb = (db: Database) => !!db.is\_audit;

REQ-1224: import { createMockDatabase } from "metabase-types/api/mocks";  
  
import { isAuditDb } from "./utils";  
  
describe("enterprise audit utils", () => {  
 describe("isAuditDb", () => {  
 it("should return...

REQ-1225: import { createAsyncThunk } from "@reduxjs/toolkit";  
  
import { redirect } from "metabase/lib/dom";  
import { getSetting } from "metabase/selectors/settings";  
import type { State } from...

REQ-1226: export \* from "./SessionTimeoutSetting";

REQ-1227: export \* from "./SettingsJWTForm";

REQ-1228: export \* from "./SettingsSAMLForm";

REQ-1229: export \* from "./SsoButton";

REQ-1230: import \* as Yup from "yup";  
  
export const JWT\_SCHEMA = Yup.object({  
 "jwt-enabled": Yup.boolean().default(false),  
 "jwt-user-provisioning-enabled?": Yup.boolean().default(null),  
 ...

REQ-1231: export \* from "./JwtAuthCard";

REQ-1232: export \* from "./SamlAuthCard";

REQ-1233: import { LOGIN, LOGIN\_GOOGLE } from "metabase/auth/actions";  
import MetabaseSettings from "metabase/lib/settings";  
import {  
 PLUGIN\_AUTH\_PROVIDERS,  
 PLUGIN\_IS\_PASSWORD\_USER,  
 ...

REQ-1234: import Cookies from "js-cookie";  
import { replace } from "react-router-redux";  
  
import { logout, refreshSession } from "metabase/auth/actions";  
import { isSameOrSiteUrlOrigin } from...

REQ-1235: import FakeTimers from "@sinonjs/fake-timers";  
import Cookie from "js-cookie";  
import { replace } from "react-router-redux";  
  
import { logout, refreshSession } from "metabase/auth/actions";  
  
import...

REQ-1236: import { useGetSettingsQuery } from "metabase/api";  
import { hasAnySsoFeature } from "metabase/common/utils/plan";  
  
export const getSSOUrl = (siteUrl: string, redirectUrl?: string): string => {  
 if...

REQ-1237: import { t } from "ttag";  
import \* as Yup from "yup";  
  
import {  
 getPerformanceTabMetadata,  
 getPositiveIntegerSchema,  
} from "metabase/admin/performance/constants/complex";  
import {  
 ...

REQ-1238: export function hasQuestionCacheSection(question) {  
 const type = question.type();  
  
 return (  
 type !== "model" &&  
 (question.canWrite() || question.lastQueryStart() != null)  
 );  
}

REQ-1239: import { useCallback, useState } from "react";  
  
export const usePagination = ({  
 pageSize: inputPageSize,  
 initialPage,  
}: {  
 pageSize: number;  
 initialPage: number;  
}) => {  
 const pageSize =...

REQ-1240: import dayjs from "dayjs";  
import { c, t } from "ttag";  
import \_ from "underscore";  
  
import type { StaleCollectionItem } from "../types";  
  
// constant portion of this string is how mantine calculates...

REQ-1241: import { trackSchemaEvent } from "metabase/lib/analytics";  
import type { RegularCollectionId } from "metabase-types/api";  
  
export const trackStaleItemsArchived = ({  
 collection\_id,  
 ...

REQ-1242: import type {  
 CollectionId,  
 CollectionItem,  
 PaginationRequest,  
 PaginationResponse,  
} from "metabase-types/api";  
import type { SortingOptions } from "metabase-types/api/sorting";  
  
export type...

REQ-1243: import {  
 isInstanceAnalyticsCustomCollection,  
 isTrashedCollection,  
} from "metabase/collections/utils";  
import type { Collection } from "metabase-types/api";  
  
export function...

REQ-1244: import { PLUGIN\_COLLECTIONS } from "metabase/plugins";  
import { createMockCollection } from "metabase-types/api/mocks";  
  
import { canCleanUp } from "./utils";  
  
describe("canCleanUp", () => {  
 ...

REQ-1245: import { t } from "ttag";  
  
import type {  
 BaseEntityId,  
 CollectionAuthorityLevelConfig,  
 CollectionInstanceAnaltyicsConfig,  
} from "metabase-types/api";  
  
export const REGULAR\_COLLECTION:...

REQ-1246: export \* from "./use-get-default-collection-id";

REQ-1247: import { skipToken, useGetCollectionQuery } from "metabase/api";  
import { \_useGetDefaultCollectionId as useOSSGetDefaultCollectionId } from "metabase/collections/hooks";  
import { useGetAuditInfoQuery...

REQ-1248: import type { IconData, ObjectWithModel } from "metabase/lib/icon";  
import { getIconBase } from "metabase/lib/icon";  
import type { ItemWithCollection } from "metabase/plugins";  
import type {  
 ...

REQ-1249: export { ContentTranslationConfiguration } from "./ContentTranslationConfiguration/ContentTranslationConfiguration";

REQ-1250: export const contentTranslationEndpoints = {  
 /\*\* This endpoint, which includes a JSON Web Token, is set in static embedding \*/  
 getDictionary: null as string | null,  
 uploadDictionary:...

REQ-1251: import { PLUGIN\_CONTENT\_TRANSLATION } from "metabase/plugins";  
import { hasPremiumFeature } from "metabase-enterprise/settings";  
  
import { ContentTranslationConfiguration } from...

REQ-1252: import type { HoveredObject } from "metabase/visualizations/types";  
import type { DatasetColumn, DictionaryArray } from "metabase-types/api";  
import { createMockColumn, createMockSeries } from...

REQ-1253: import { useCallback } from "react";  
  
import { skipToken } from "metabase/api";  
import { useLocale } from "metabase/common/hooks";  
import type { ContentTranslationFunction } from...

REQ-1254: import \* as I from "icepick";  
import { useCallback, useMemo } from "react";  
import \_ from "underscore";  
  
import type { ContentTranslationFunction } from "metabase/i18n/types";  
import type {...

REQ-1255: export \* from "./VerifiedFilter";

REQ-1256: import { PLUGIN\_CONTENT\_VERIFICATION } from "metabase/plugins";  
import { hasPremiumFeature } from "metabase-enterprise/settings";  
  
import { VerifiedFilter } from "./VerifiedFilter";  
import {...

REQ-1257: import \_ from "underscore";  
  
import type { Database } from "metabase-types/api";  
  
const FILTER\_DB\_KEYS = ["id", "name"];  
  
const FILTER\_DB\_DETAIL\_KEYS = [  
 "password",  
 "schema-filters-type",  
 ...

REQ-1258: import { useEffect } from "react";  
import { replace } from "react-router-redux";  
  
import { useDispatch } from "metabase/lib/redux";  
import \* as Urls from "metabase-enterprise/urls";  
import type {...

REQ-1259: import { skipToken } from "@reduxjs/toolkit/query";  
  
export function paramIdToGetQuery(strId: string | undefined) {  
 const id = strId ? parseInt(strId, 10) : undefined;  
 return id ? { id } :...

REQ-1260: export { EmbeddingAppOriginDescription } from "./EmbeddingAppOriginDescription";

REQ-1261: export { SameSiteSelectWidget } from "./SameSiteSelectWidget";

REQ-1262: import { DataSourceSelector } from "embedding/data-picker/DataSelector";  
import { SimpleDataPicker } from "embedding/data-picker/SimpleDataPicker";  
import { PLUGIN\_ADMIN\_SETTINGS, PLUGIN\_EMBEDDING }...

REQ-1263: import { getPlan } from "metabase/common/utils/plan";  
import { getSetting } from "metabase/selectors/settings";  
import type { EnterpriseState } from "metabase-enterprise/settings/types";  
  
export...

REQ-1264: import { PLUGIN\_EMBEDDING\_SDK } from "metabase/plugins";  
import { hasPremiumFeature } from "metabase-enterprise/settings";  
  
if (hasPremiumFeature("embedding\_sdk")) {  
 PLUGIN\_EMBEDDING\_SDK.isEnabled...

REQ-1265: import type {  
 DashboardEmbedOptions,  
 ExplorationEmbedOptions,  
 QuestionEmbedOptions,  
 SdkIframeEmbedBaseSettings,  
 SdkIframeEmbedSettingKey,  
} from "./types/embed";  
  
/\*\*  
 \* The timeout to wait...

REQ-1266: import {  
 connectToInstanceAuthSso,  
 jwtDefaultRefreshTokenFunction,  
 openSamlLoginPopup,  
 validateSessionToken,  
} from "embedding/auth-common";  
import { INVALID\_AUTH\_METHOD, MetabaseError } from...

REQ-1267: import { act } from "react-dom/test-utils";  
  
import type { MetabaseEmbedElement } from "./embed";  
  
const defineMetabaseConfig = (config: unknown) => {  
 (window as any).metabaseConfig =...

REQ-1268: import { useMemo } from "react";  
import { P, match } from "ts-pattern";  
  
import { sortObject } from "metabase-lib/v1/utils";  
  
import type { SdkIframeEmbedSettings } from...

REQ-1269: import { useEffect, useState } from "react";  
import { match } from "ts-pattern";  
  
import { isWithinIframe } from "metabase/lib/dom";  
  
import type {  
 SdkIframeEmbedMessage,  
 ...

REQ-1270: import { PLUGIN\_EMBEDDING\_IFRAME\_SDK } from "metabase/plugins";  
import { hasPremiumFeature } from "metabase-enterprise/settings";  
  
import { SdkIframeEmbedRoute } from...

REQ-1271: import type {  
 EntityTypeFilterKeys,  
 MetabaseTheme,  
 SqlParameterValues,  
} from "embedding-sdk";  
import type { MetabaseError } from "embedding-sdk/errors";  
import type { MetabaseAuthMethod } from...

REQ-1272: import type { Action, Store } from "@reduxjs/toolkit";  
  
import type { SdkStoreState } from "embedding-sdk/store/types";  
  
export type StoreWithSdkState = Store<SdkStoreState, Action>;

REQ-1273: export { requestSessionTokenFromEmbedJs } from "./request-session-token";

REQ-1274: import { samlTokenStorage } from "embedding/auth-common";  
import { AUTH\_TIMEOUT } from "embedding-sdk/errors";  
import type { MetabaseEmbeddingSessionToken } from...

REQ-1275: import json5 from "json5";  
  
// Convert kebab-case attribute names to their camelCase setting keys.  
export const attributeToSettingKey = (attr: string): string =>  
 attr.replace(/-([a-z])/g, (\_, c) =>...

REQ-1276: import { parseAttributeValue } from "./webcomponents";  
  
// Note, these tests use ` when to wrap strings passed to parseAttributeValue so we can only focus on ' vs " inside the...

REQ-1277: import { trackSimpleEvent } from "metabase/lib/analytics";  
import type {  
 SdkIframeEmbedSettingKey,  
 SdkIframeEmbedSettings,  
} from "metabase-enterprise/embedding\_iframe\_sdk/types/embed";  
  
import...

REQ-1278: import { useSdkIframeEmbedSetupContext } from "../../../context";  
  
export function useHideParameter() {  
 const { settings, updateSettings } = useSdkIframeEmbedSetupContext();  
  
 const...

REQ-1279: import { match } from "ts-pattern";  
  
import type { Parameter, ParameterType } from "metabase-types/api";  
  
export const getParameterPlaceholder = (param: Parameter): string => {  
 // If the parameter...

REQ-1280: import type { Parameter } from "metabase-types/api";  
  
import { getParameterPlaceholder } from "./parameter-placeholder";  
  
const createParameter = (overrides: Partial<Parameter> = {}): Parameter =>...

REQ-1281: import { GetCodeStep } from "./components/GetCodeStep";  
import { SelectEmbedExperienceStep } from "./components/SelectEmbedExperienceStep";  
import { SelectEmbedOptionsStep } from...

REQ-1282: import { createContext, useContext } from "react";  
  
import type { Parameter } from "metabase-types/api";  
  
import type {  
 SdkIframeEmbedSetupExperience,  
 SdkIframeEmbedSetupRecentItem,  
 ...

REQ-1283: export { useSdkIframeEmbedNavigation } from "./use-sdk-iframe-embed-navigation";  
export { useRecentItems } from "./use-recent-items";  
export { useParameterList } from "./use-parameter-list";

REQ-1284: import { useMemo } from "react";  
import { useLatest } from "react-use";  
  
import { skipToken, useGetCardQuery, useGetDashboardQuery } from "metabase/api";  
import { useSelector } from...

REQ-1285: import { useCallback, useMemo, useState } from "react";  
  
import { useListRecentsQuery } from "metabase/api";  
import type { RecentItem } from "metabase-types/api";  
  
import {...

REQ-1286: import { useMemo } from "react";  
  
import { EMBED\_STEPS } from "../constants";  
import { useSdkIframeEmbedSetupContext } from "../context";  
  
export function useSdkIframeEmbedNavigation() {  
 const {...

REQ-1287: import { useMemo } from "react";  
  
import { useSetting } from "metabase/common/hooks";  
  
import { useSdkIframeEmbedSetupContext } from "../context";  
import { getEmbedSnippet } from...

REQ-1288: import { PLUGIN\_EMBEDDING\_IFRAME\_SDK\_SETUP } from "metabase/plugins";  
import { hasPremiumFeature } from "metabase-enterprise/settings";  
  
import { SdkIframeEmbedSetup } from...

REQ-1289: import type {  
 SdkIframeEmbedBaseSettings,  
 SdkIframeEmbedTemplateSettings,  
} from "metabase-enterprise/embedding\_iframe\_sdk/types/embed";  
import type { BaseRecentItem } from...

REQ-1290: import { match } from "ts-pattern";  
  
import type {  
 DashboardEmbedOptions,  
 ExplorationEmbedOptions,  
 QuestionEmbedOptions,  
} from "metabase-enterprise/embedding\_iframe\_sdk/types/embed";  
  
import...

REQ-1291: import { match } from "ts-pattern";  
import \_ from "underscore";  
  
import {  
 ALLOWED\_EMBED\_SETTING\_KEYS\_MAP,  
 type AllowedEmbedSettingKey,  
} from...

REQ-1292: \*

REQ-1293: import { t } from "ttag";  
  
import type { MetabaseColors } from "embedding-sdk";  
import type { ColorName } from "metabase/lib/colors/types";  
  
export const getConfigurableThemeColors = () =>  
 [  
 {  
...

REQ-1294: import {  
 PLUGIN\_ADMIN\_ALLOWED\_PATH\_GETTERS,  
 PLUGIN\_FEATURE\_LEVEL\_PERMISSIONS,  
} from "metabase/plugins";  
import { hasPremiumFeature } from "metabase-enterprise/settings";  
  
import {...

REQ-1295: import { push } from "react-router-redux";  
import { t } from "ttag";  
  
import { UNABLE\_TO\_CHANGE\_ADMIN\_PERMISSIONS } from "metabase/admin/permissions/constants/messages";  
import {  
 ...

REQ-1296: import { DataPermissionValue } from "metabase/admin/permissions/types";  
import type { Group, GroupsPermissions } from "metabase-types/api";  
  
import {  
 DATA\_MODEL\_PERMISSION\_OPTIONS,  
 ...

REQ-1297: import { getIn } from "icepick";  
import { t } from "ttag";  
  
import { UNABLE\_TO\_CHANGE\_ADMIN\_PERMISSIONS } from "metabase/admin/permissions/constants/messages";  
import {  
 getPermissionWarning,  
 ...

REQ-1298: import { DataPermissionValue } from "metabase/admin/permissions/types";  
import type { Group, GroupsPermissions } from "metabase-types/api";  
  
import {  
 DETAILS\_PERMISSION\_OPTIONS,  
 ...

REQ-1299: import { push } from "react-router-redux";  
import { t } from "ttag";  
  
import { UNABLE\_TO\_CHANGE\_ADMIN\_PERMISSIONS } from "metabase/admin/permissions/constants/messages";  
import {  
 ...

REQ-1300: import { DataPermissionValue } from "metabase/admin/permissions/types";  
import type { Group, GroupsPermissions } from "metabase-types/api";  
  
import {  
 DOWNLOAD\_PERMISSION\_OPTIONS,  
 ...

REQ-1301: import type {  
 DataPermissionValue,  
 EntityId,  
 PermissionSubject,  
} from "metabase/admin/permissions/types";  
import type { Group, GroupsPermissions } from "metabase-types/api";  
  
import {...

REQ-1302: import { t } from "ttag";  
  
import type { Dataset } from "metabase-types/api/dataset";  
  
export const canDownloadResults = (result: Dataset) =>  
 result.data?.download\_perms !== "none";  
  
export const...

REQ-1303: import type { User } from "metabase-types/api";  
  
export interface UserWithFeaturePermissions extends User {  
 permissions?: {  
 can\_access\_data\_model: boolean;  
 can\_access\_db\_details: boolean;  
 ...

REQ-1304: import { t } from "ttag";  
  
import type { PermissionSubject } from "metabase/admin/permissions/types";  
import type { AdminPathKey } from "metabase-types/store";  
  
import type {...

REQ-1305: import { t } from "ttag";  
  
export const getStrings = (objectType: "file" | "folder") => {  
 if (objectType === "file") {  
 return {  
 clickInstruction: t`In Google Drive, right-click on the...

REQ-1306: import { trackSimpleEvent } from "metabase/lib/analytics";  
import type { GsheetsConnectionClickedEvent } from "metabase-types/analytics";  
  
export function trackSheetImportClick() {  
 ...

REQ-1307: export const SYNC\_POLL\_INTERVAL = 3 \* 1000; // 3 seconds

REQ-1308: export \* from "./GdriveAddDataPanel";  
export \* from "./GdriveConnectionModal";  
export \* from "./GdriveSyncStatus";  
export \* from "./GdriveDbMenu";  
export \* from "./PausedModal";

REQ-1309: import { getStatus } from "./utils";  
  
describe("google\_drive > getStatus", () => {  
 it("should return 'not-connected' if status is undefined", () => {  
 const result = getStatus({ status:...

REQ-1310: import { push } from "react-router-redux";  
  
import { getAdminPaths } from "metabase/admin/app/selectors";  
import { permissionApi } from "metabase/api";  
import { createThunkAction } from...

REQ-1311: export \* from "./UserTypeCell";

REQ-1312: export \* from "./UserTypeToggle";

REQ-1313: import {  
 PLUGIN\_ADMIN\_ALLOWED\_PATH\_GETTERS,  
 PLUGIN\_GROUP\_MANAGERS,  
} from "metabase/plugins";  
import { hasPremiumFeature } from "metabase-enterprise/settings";  
  
import {  
 ...

REQ-1314: import type { User } from "metabase-types/api";  
  
export interface UserWithGroupManagerPermission extends User {  
 permissions?: {  
 is\_group\_manager: boolean;  
 };  
}

REQ-1315: import { t } from "ttag";  
  
import type { ConfirmationState } from "metabase/common/hooks/use-confirmation";  
import type { Member, Membership } from "metabase-types/api";  
import type { User } from...

REQ-1316: import { formatDateTimeWithUnit, formatNumber } from "metabase/lib/formatting";  
import type { BillingInfoLineItem } from "metabase-types/api";  
import {  
 supportedDisplayTypes,  
 ...

REQ-1317: export \* from "./LicenseAndBillingSettings";

REQ-1318: export \* from "./StillNeedHelp";

REQ-1319: import { PLUGIN\_ADMIN\_SETTINGS } from "metabase/plugins";  
  
import { LicenseAndBillingSettings } from "./components/LicenseAndBillingSettings";  
import { useUpsellFlow } from...

REQ-1320: import { useEffect, useRef } from "react";  
import { t } from "ttag";  
  
import { getCurrentUser } from "metabase/admin/datamodel/selectors";  
import { useUpsellLink } from...

REQ-1321: Metabot v3  
# Sending requests to Metabot  
## Request format

REQ-1322: import type { MetabotId, SuggestedMetabotPrompt } from "metabase-types/api";  
  
export const mockSuggestedPrompts: Record<MetabotId, SuggestedMetabotPrompt[]> =  
 {  
 "1": [  
 {  
 id: 1,  
 ...

REQ-1323: import { useSelector } from "metabase/lib/redux";  
import { getLocation } from "metabase/selectors/routing";  
  
export function useMetabotIdPath() {  
 const location = useSelector(getLocation);  
 const...

REQ-1324: import { type RefObject, useCallback, useEffect, useRef } from "react";  
  
function calculateFillerHeight(  
 scrollContainerEl: HTMLElement,  
 fillerEl: HTMLElement,  
): number {  
 const scrollContent =...

REQ-1325: export \* from "./MetabotIcon";

REQ-1326: import { t } from "ttag";  
  
export const LONG\_CONVO\_MSG\_LENGTH\_THRESHOLD = 120000;  
  
// NOTE: this is not ideal, but will get fixed w/ BOT-189 allowing us to use fixed entity\_ids  
export const...

REQ-1327: import { isFulfilled } from "@reduxjs/toolkit";  
import { useCallback } from "react";  
  
import { useDispatch, useSelector } from "metabase/lib/redux";  
import { useMetabotContext } from...

REQ-1328: import { t } from "ttag";  
  
import { addAgentErrorMessage, stopProcessing } from "../state";  
  
import type { ReactionHandler } from "./types";  
  
type UnknownReaction = { type: string } & Record<string,...

REQ-1329: import type { MetabotReaction } from "metabase-types/api";  
  
import { showMessage } from "./messages";  
import { redirect } from "./metabot";  
import type { ReactionHandler } from "./types";  
  
export \*...

REQ-1330: import type { MetabotMessageReaction } from "metabase-types/api";  
  
import { addAgentMessage } from "../state";  
  
import type { ReactionHandler } from "./types";  
  
export const showMessage:...

REQ-1331: import { push } from "react-router-redux";  
  
import type { MetabotRedirectReaction } from "metabase-types/api";  
  
import type { ReactionHandler } from "./types";  
  
export const redirect:...

REQ-1332: import type { Dispatch, GetState } from "metabase-types/store";  
  
export type ReactionHandler<Reaction> = (  
 reaction: Reaction,  
) => (reduxApis: {  
 dispatch: Dispatch;  
 getState: GetState;  
}) =>...

REQ-1333: import { type UnknownAction, isRejected } from "@reduxjs/toolkit";  
import { push } from "react-router-redux";  
import { P, match } from "ts-pattern";  
  
import { createAsyncThunk } from...

REQ-1334: export \* from "./actions";  
export \* from "./reducer";  
export \* from "./selectors";  
export \* from "./types";

REQ-1335: import { type PayloadAction, createSlice } from "@reduxjs/toolkit";  
import \_ from "underscore";  
  
import { logout } from "metabase/auth/actions";  
import { uuid } from "metabase/lib/uuid";  
import type...

REQ-1336: import { createSelector } from "@reduxjs/toolkit";  
import \_ from "underscore";  
  
import { getIsEmbedding } from "metabase/selectors/embed";  
  
import {  
 FIXED\_METABOT\_IDS,  
 ...

REQ-1337: import { setupEnterprisePlugins } from "\_\_support\_\_/enterprise";  
import { createMockState } from "metabase-types/store/mocks";  
  
import {  
 type MetabotState,  
 getLastAgentMessagesByType,  
 ...

REQ-1338: import type { EnterpriseSharedState } from "metabase-enterprise/shared/reducer";  
import type { EnterpriseState } from "metabase-enterprise/shared/types";  
  
import type { MetabotState } from...

REQ-1339: import { nanoid } from "@reduxjs/toolkit";  
  
export const createMessageId = () => {  
 return `msg\_${nanoid()}`;  
};

REQ-1340: export \* from "./ModelCacheControl";

REQ-1341: import { PLUGIN\_MODEL\_PERSISTENCE } from "metabase/plugins";  
import { hasPremiumFeature } from "metabase-enterprise/settings";  
  
import { ModelCacheToggle } from "./components/ModelCacheControl";  
  
if...

REQ-1342: export \* from "./EntityModerationIcon";

REQ-1343: export \* from "./ModerationReviewBanner";

REQ-1344: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1345: export \* from "./ModerationStatusIcon";

REQ-1346: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1347: import type { ColorName } from "metabase/lib/colors/types";  
import type { IconName } from "metabase/ui";  
  
export const MODERATION\_STATUS = {  
 verified: "verified",  
};  
  
export const...

REQ-1348: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1349: import { PLUGIN\_MODERATION } from "metabase/plugins";  
import { hasPremiumFeature } from "metabase-enterprise/settings";  
  
import { EntityModerationIcon } from...

REQ-1350: import { getUserIsAdmin } from "metabase/selectors/user";  
  
export const getIsModerator = (state, props) => {  
 return getUserIsAdmin(state, props);  
};

REQ-1351: import { c, t } from "ttag";  
import \_ from "underscore";  
  
import type { ColorName } from "metabase/lib/colors/types";  
import { ModerationReviewApi } from "metabase/services";  
import type { IconName }...

REQ-1352: import { ModerationReviewApi } from "metabase/services";  
import type Question from "metabase-lib/v1/Question";  
import type { ModerationReview } from "metabase-types/api";  
import {  
 ...

REQ-1353: import { t } from "ttag";  
  
import type Question from "metabase-lib/v1/Question";  
  
export const getVerifyQuestionTitle = (question: Question): string => {  
 const type = question.type();  
  
 if (type...

REQ-1354: import "./whitelabel/overrides";

REQ-1355: import { PLUGIN\_IS\_EE\_BUILD } from "metabase/plugins";  
  
// SETTINGS OVERRIDES:  
PLUGIN\_IS\_EE\_BUILD.isEEBuild = () => true;  
  
import "./shared";  
  
// PLUGINS:  
  
import "./tools";  
import...

REQ-1356: import type { TypedUseSelectorHook } from "react-redux";  
import { createSelectorHook } from "react-redux";  
  
import { MetabaseReduxContext } from "metabase/lib/redux";  
  
// TODO: use the real type...

REQ-1357: import "./resource\_downloads\_plugin";

REQ-1358: import { P, match } from "ts-pattern";  
  
import { PLUGIN\_RESOURCE\_DOWNLOADS } from "metabase/plugins";  
import type { EmbedResourceDownloadOptions } from "metabase/public/lib/types";  
import {...

REQ-1359: import { PLUGIN\_RESOURCE\_DOWNLOADS } from "metabase/plugins";  
  
import { downloadsEnabledTestData, setup } from "./setup";  
  
describe("[OSS] resource downloads plugin", () => {  
 beforeAll(() => {  
 ...

REQ-1360: import { PLUGIN\_RESOURCE\_DOWNLOADS } from "metabase/plugins";  
  
import { downloadsEnabledTestData, setup } from "./setup";  
  
describe("[EE - no features] resource downloads plugin", () => {  
 ...

REQ-1361: import { PLUGIN\_RESOURCE\_DOWNLOADS } from "metabase/plugins";  
import { createMockTokenFeatures } from "metabase-types/api/mocks";  
  
import { downloadsEnabledTestData, setup } from...

REQ-1362: import { setupEnterprisePlugins } from "\_\_support\_\_/enterprise";  
import { mockSettings } from "\_\_support\_\_/settings";  
import type { EmbedResourceDownloadOptions } from...

REQ-1363: import \_ from "underscore";  
  
import {  
 LOAD\_DATA\_PERMISSIONS,  
 SAVE\_DATA\_PERMISSIONS,  
 UPDATE\_DATA\_PERMISSION,  
 updateDataPermission,  
} from "metabase/admin/permissions/permissions";  
import {  
 ...

REQ-1364: export \* from "./DataAttributeMappingEditor";  
export \* from "./AttributeOptionsEmptyState";

REQ-1365: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1366: export \* from "./LoginAttributesWidget";

REQ-1367: import { t } from "ttag";  
  
import {  
 DataPermissionValue,  
 type EntityId,  
} from "metabase/admin/permissions/types";  
import { hasPermissionValueInGraph } from...

REQ-1368: import { push } from "react-router-redux";  
import { t } from "ttag";  
import \_ from "underscore";  
  
import { DataPermissionValue } from "metabase/admin/permissions/types";  
import {  
 ...

REQ-1369: import { createSelector } from "@reduxjs/toolkit";  
  
import type { GroupTableAccessPolicyParams, SandboxesState } from "./types";  
import { getPolicyKeyFromParams } from "./utils";  
  
export const...

REQ-1370: import type { EnterpriseSharedState } from "metabase-enterprise/shared/reducer";  
import type { EnterpriseState } from "metabase-enterprise/shared/types";  
import type {  
 DimensionRef,  
 ...

REQ-1371: \*

REQ-1372: import { PLUGIN\_IS\_EE\_BUILD } from "metabase/plugins";  
  
// SETTINGS OVERRIDES:  
PLUGIN\_IS\_EE\_BUILD.isEEBuild = () => true;  
  
import "./shared";  
  
// SDK PLUGINS:  
  
import "./embedding";  
import...

REQ-1373: import { DELETE, GET } from "metabase/lib/api";  
  
export const AuditApi = {  
 unsubscribe\_user: DELETE("/api/ee/audit-app/user/:id/subscriptions"),  
};  
  
export const ImpersonationApi = {  
 get:...

REQ-1374: import { useCallback, useEffect, useState } from "react";  
import { t } from "ttag";  
  
import { reload } from "metabase/lib/dom";  
import { SettingsApi, StoreApi } from "metabase/services";  
import type...

REQ-1375: import { isTest } from "metabase/env";  
import MetabaseSettings from "metabase/lib/settings";  
import type { TokenFeature } from "metabase-types/api";  
  
export function hasPremiumFeature(feature:...

REQ-1376: import noResultsSource from "assets/img/no\_results.svg";  
import type { IllustrationValue } from "metabase/plugins";  
import { getSetting, getSettings } from "metabase/selectors/settings";  
import type...

REQ-1377: import {  
 createMockSettingsState,  
 createMockState,  
} from "metabase-types/store/mocks";  
  
import { getIsWhiteLabeling, getLoadingMessage, getLogoUrl } from "./selectors";  
  
describe("getLogoUrl",...

REQ-1378: import type { EnterpriseSettings } from "metabase-types/api";  
import type { SettingsState, State } from "metabase-types/store";  
  
export interface EnterpriseState extends State {  
 settings:...

REQ-1379: import { PLUGIN\_REDUCERS } from "metabase/plugins";  
  
import { shared } from "./reducer";  
  
PLUGIN\_REDUCERS.shared = shared.reducer;

REQ-1380: import { createSlice } from "@reduxjs/toolkit";  
  
import { createAsyncThunk } from "metabase/lib/redux";  
import { GTAPApi } from "metabase/services";  
import type { UserAttributeKey } from...

REQ-1381: import type { EnterpriseState } from "./types";  
  
export const getUserAttributes = (state: EnterpriseState) => {  
 return state.plugins.shared.attributes;  
};

REQ-1382: import type { State } from "metabase-types/store";  
  
import type { EnterpriseSharedState } from "./reducer";  
  
export interface EnterpriseState extends State {  
 plugins: {  
 shared:...

REQ-1383: import { hasInlineParameters } from "metabase/dashboard/utils";  
import type { UiParameter } from "metabase-lib/v1/parameters/types";  
import type { Dashboard, DashboardCard } from...

REQ-1384: import { PLUGIN\_DASHBOARD\_SUBSCRIPTION\_PARAMETERS\_SECTION\_OVERRIDE } from "metabase/plugins";  
import { hasPremiumFeature } from "metabase-enterprise/settings";  
import { MutableParametersSection }...

REQ-1385: export { CloudSMTPConnectionCard } from "./CloudSMTPConnectionCard";  
export { SMTPOverrideConnectionForm } from "./SMTPOverrideConnectionForm";

REQ-1386: import { PLUGIN\_SMTP\_OVERRIDE } from "metabase/plugins";  
import { hasPremiumFeature } from "metabase-enterprise/settings";  
  
import {  
 CloudSMTPConnectionCard,  
 SMTPOverrideConnectionForm,  
} from...

REQ-1387: import { t } from "ttag";  
  
import CollectionPermissionsModal from "metabase/admin/permissions/components/CollectionPermissionsModal/CollectionPermissionsModal";  
import { canonicalCollectionId } from...

REQ-1388: import { applyWhitelabelOverride } from "./whitelabel/static-viz-overrides";  
  
export default function apply() {  
 applyWhitelabelOverride();  
}

REQ-1389: import { PLUGIN\_ADMIN\_TOOLS } from "metabase/plugins";  
import { hasPremiumFeature } from "metabase-enterprise/settings";  
  
import ErrorOverview from "./ErrorOverview";  
  
if...

REQ-1390: import { push } from "react-router-redux";  
import { t } from "ttag";  
  
const CARD\_ID\_ROW\_IDX = 0;  
  
const ErrorDrill = ({ clicked }) => {  
 if (!clicked) {  
 return [];  
 }  
  
 const cardId =...

REQ-1391: import { PLUGIN\_UPLOAD\_MANAGEMENT } from "metabase/plugins";  
import { hasPremiumFeature } from "metabase-enterprise/settings";  
  
import {  
 FileUploadErrorModal,  
 GdriveAddDataPanel,  
 ...

REQ-1392: import dayjs from "dayjs";  
import relativeTime from "dayjs/plugin/relativeTime";  
import updateLocale from "dayjs/plugin/updateLocale";  
dayjs.extend(updateLocale);  
dayjs.extend(relativeTime);  
  
export...

REQ-1393: import type { DatabaseId } from "metabase-types/api";  
  
export \* as Urls from "metabase/lib/urls";  
  
export function viewDestinationDatabases(databaseId: DatabaseId) {  
 return...

REQ-1394: import { PLUGIN\_AUTH\_PROVIDERS } from "metabase/plugins";  
import { hasPremiumFeature } from "metabase-enterprise/settings";  
  
import { UserProvisioning } from "./components/UserProvisioning";  
  
if...

REQ-1395: export interface MaskedScimApiKey {  
 id: number;  
 scope: "scim";  
 key: string;  
 key\_prefix: string;  
 masked\_key: string;  
 name: string;  
 user\_id: null;  
 created\_at: string;  
 creator\_id:...

REQ-1396: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1397: export interface ColorOption {  
 name: string;  
 description: string;  
}

REQ-1398: import { t } from "ttag";  
  
import type { ColorOption } from "./types";  
  
export const getBrandColorOptions = (): ColorOption[] => [  
 {  
 name: "brand",  
 description: t`The main color used...

REQ-1399: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1400: import \_ from "underscore";  
  
import { color } from "metabase/lib/colors";  
  
export const getAccentColorGroups = (palette: Record<string, string>) => {  
 const groups = [  
 \_.times(8, (i) =>...

REQ-1401: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1402: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1403: import Color from "color";  
import \_ from "underscore";  
  
export const getChartColorGroups = (): string[][] => {  
 return \_.times(8, (i) => [  
 `accent${i}`,  
 `accent${i}-light`,  
 ...

REQ-1404: import Color from "color";  
  
import { getAutoChartColors, getDefaultChartColors } from "./utils";  
  
describe("getDefaultChartColors", () => {  
 const groups = [["accent1"], ["accent2"], ["accent3"]];  
  
...

REQ-1405: export { ColorSettings } from "./ColorSettings";

REQ-1406: export \* from "./ColorSettingsWidget";

REQ-1407: export \* from "./FontWidget";

REQ-1408: import type { FontFile, Settings } from "metabase-types/api";  
  
export interface FontSetting {  
 value: string | null;  
 default: string;  
}  
  
export type FontSettingKeys = "application-font" |...

REQ-1409: import { useMemo } from "react";  
import { t } from "ttag";  
import \_ from "underscore";  
  
import { useAdminSetting } from "metabase/api/utils";  
import type { FontFile, FontFormat } from...

REQ-1410: export { HelpLinkSettings } from "./HelpLinkSettings";

REQ-1411: export { IllustrationWidget } from "./IllustrationWidget";

REQ-1412: export { ImageToggle } from "./ImageToggle";

REQ-1413: interface GetRelativeLandingPageUrlResult {  
 isSameOrigin: boolean;  
 relativeUrl: string;  
}  
  
export const getRelativeLandingPageUrl = (  
 value: string,  
): GetRelativeLandingPageUrlResult => {  
 ...

REQ-1414: import { getRelativeLandingPageUrl } from "./utils";  
  
describe("utils", () => {  
 describe("getRelativeLandingPageUrl", () => {  
 it("should return relative url for valid inputs", () => {  
 [  
 ...

REQ-1415: export { MetabotToggleWidget } from "./MetabotToggleWidget";

REQ-1416: import MetabaseSettings from "metabase/lib/settings";  
import {  
 PLUGIN\_APP\_INIT\_FUNCTIONS,  
 PLUGIN\_LANDING\_PAGE,  
 PLUGIN\_LOGO\_ICON\_COMPONENTS,  
 PLUGIN\_SELECTORS,  
 PLUGIN\_WHITELABEL,  
} from...

REQ-1417: import { t } from "ttag";  
  
export const LOADING\_MESSAGE\_BY\_SETTING = {  
 "doing-science": {  
 // eslint-disable-next-line ttag/no-module-declaration -- see metabase#55045  
 name: t`Doing...

REQ-1418: import { colors } from "metabase/lib/colors/colors";  
import MetabaseSettings from "metabase/lib/settings";  
  
export function updateColors() {  
 const scheme =...

REQ-1419: import { hasPremiumFeature } from "metabase-enterprise/settings";  
import { updateColors } from "metabase-enterprise/whitelabel/lib/whitelabel";  
  
if (hasPremiumFeature("whitelabel")) {  
 ...

REQ-1420: import { hasPremiumFeature } from "metabase-enterprise/settings";  
import { updateColors } from "metabase-enterprise/whitelabel/lib/whitelabel";  
  
export function applyWhitelabelOverride() {  
 if...

REQ-1421: Working in /frontend  
# Structure

REQ-1422: \*

REQ-1423: \*

REQ-1424: eslint-disable import/no-commonjs  
eslint-env node

REQ-1425: \*

REQ-1426: ------------------------------------------------------------------------------  
Rule Definition  
------------------------------------------------------------------------------  
eslint-disable-next-line...

REQ-1427: ------------------------------------------------------------------------------  
Rule Definition  
------------------------------------------------------------------------------  
  
The following cases are...

REQ-1428: \*

REQ-1429: eslint-disable-next-line import/no-commonjs

REQ-1430: ------------------------------------------------------------------------------  
Rule Definition  
------------------------------------------------------------------------------

REQ-1431: import { RuleTester } from "eslint";  
  
import noLiteralMetabaseString from "../eslint-rules/no-literal-metabase-strings";  
  
const ruleTester = new RuleTester({  
 parserOptions: {  
 ecmaVersion:...

REQ-1432: import { RuleTester } from "eslint";  
  
import noLocaleWithIntlFunctions from "../eslint-rules/no-locale-with-intl-functions";  
  
const ruleTester = new RuleTester({  
 parserOptions: {  
 ecmaVersion:...

REQ-1433: import { RuleTester } from "eslint";  
  
import noUnconditionalMetabaseLinksRender from "../eslint-rules/no-unconditional-metabase-links-render";  
  
const ruleTester = new RuleTester({  
 parserOptions: {  
...

REQ-1434: import { RuleTester } from "eslint";  
  
import rule from "../eslint-rules/no-unordered-test-helpers";  
  
const ruleTester = new RuleTester({ parserOptions: { ecmaVersion: 2015 } });  
  
const orderError =...

REQ-1435: import { RuleTester } from "eslint";  
  
import rule from "../eslint-rules/no-unsafe-element-filtering";  
  
const ruleTester = new RuleTester({ parserOptions: { ecmaVersion: 2015 } });  
  
const unsafeError...

REQ-1436: import { RuleTester } from "eslint";  
  
import rule from "../eslint-rules/no-unscoped-text-selectors";  
  
const ruleTester = new RuleTester({ parserOptions: { ecmaVersion: 2015 } });  
  
const scopeError =...

REQ-1437: !/usr/bin/env node  
eslint-disable import/no-commonjs, no-undef, no-console

REQ-1438: {  
 "rules": {  
 // Note: adding this rule to a eslint config file in a subfolder will remove  
 // \*not\* carry over the restricted imports from parent folders, you will  
 // need to copy them...

REQ-1439: export { AccountHeader } from "./AccountHeader";

REQ-1440: import { render, screen } from "\_\_support\_\_/ui";  
  
import AccountLayout from "./AccountLayout";  
  
const getUser = () => ({  
 id: 1,  
 first\_name: "John",  
 last\_name: "Doe",  
 email:...

REQ-1441: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1442: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1443: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1444: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1445: import { push } from "react-router-redux";  
  
const PREFIX = `/account/notifications`;  
  
const TYPE\_MAP = {  
 "question-notification": "alert",  
 pulse: "pulse",  
};  
  
export const navigateToUnsubscribe =...

REQ-1446: import { render, screen, waitFor } from "\_\_support\_\_/ui";  
  
import ArchiveModal from "./ArchiveModal";  
  
const getAlert = ({ creator = getUser(), channels = [getChannel()] } = {}) => ({  
 creator,  
 ...

REQ-1447: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1448: import { mockSettings } from "\_\_support\_\_/settings";  
import { renderWithProviders, screen } from "\_\_support\_\_/ui";  
  
import HelpModal from "./HelpModal";  
  
function setup({ adminEmail, onClose } = {})...

REQ-1449: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1450: export { DashboardNotificationCard } from "./DashboardNotificationCard";  
export { NotificationCard } from "./NotificationCard";

REQ-1451: import { t } from "ttag";  
  
import type { NotificationListItem } from "metabase/account/notifications/types";  
import { formatDateTimeWithUnit } from "metabase/lib/formatting/date";  
import Settings...

REQ-1452: export { NotificationList } from "./NotificationList";

REQ-1453: import { render, screen, waitFor } from "\_\_support\_\_/ui";  
  
import UnsubscribeModal from "./UnsubscribeModal";  
  
const getAlert = ({ creator = getUser({ id: 1 }) } = {}) => ({  
 name: "Alert",  
 ...

REQ-1454: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1455: export { DeleteAlertModal } from "./DeleteAlertModal";

REQ-1456: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1457: export { NotificationsApp } from "./NotificationsApp";

REQ-1458: export { UnsubscribeAlertModal } from "./UnsubscribeAlertModal";

REQ-1459: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1460: export const getAlertId = (rawAlertId: unknown): number | null => {  
 if (rawAlertId && typeof rawAlertId === "string") {  
 return parseInt(rawAlertId, 10);  
 }  
  
 return null;  
};  
  
export const...

REQ-1461: import type { Alert, Notification } from "metabase-types/api";  
  
export type DashboardAlertListItem = {  
 item: Alert;  
 type: "pulse";  
};  
  
export type QuestionNotificationListItem = {  
 item:...

REQ-1462: import { getIn } from "icepick";  
  
import MetabaseSettings from "metabase/lib/settings";  
import { UtilApi } from "metabase/services";  
  
export const validatePassword = async (password: string) => {  
 ...

REQ-1463: export \* from "./UserPasswordForm";

REQ-1464: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1465: export interface UserPasswordData {  
 old\_password: string;  
 password: string;  
 password\_confirm: string;  
}

REQ-1466: import { userApi } from "metabase/api";  
import { createThunkAction } from "metabase/lib/redux";  
import type { User } from "metabase-types/api";  
  
import type { UserProfileData } from...

REQ-1467: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1468: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1469: import { createSelector } from "@reduxjs/toolkit";  
  
import { PLUGIN\_IS\_PASSWORD\_USER } from "metabase/plugins";  
import { getSettings } from "metabase/selectors/settings";  
import { getUser } from...

REQ-1470: export interface UserProfileData {  
 first\_name?: string | null;  
 last\_name?: string | null;  
 email?: string;  
 locale: string | null;  
}

REQ-1471: import { addUndo } from "metabase/redux/undo";  
import { ActionsApi } from "metabase/services";  
import type {  
 ActionFormSubmitResult,  
 ParametersForActionExecution,  
 WritebackAction,  
} from...

REQ-1472: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1473: export { ActionFormFieldWidget } from "./ActionFormFieldWidget";

REQ-1474: export { ExplainerText } from "./ExplainerText";

REQ-1475: import { screen } from "\_\_support\_\_/ui";  
  
import { setup } from "./setup";  
  
describe("ExplainerText (OSS)", () => {  
 it("should render help link when `show-metabase-links: true`", () => {  
 ...

REQ-1476: import { screen } from "\_\_support\_\_/ui";  
  
import type { SetupOpts } from "./setup";  
import { setup as baseSetup } from "./setup";  
  
function setup(opts: SetupOpts) {  
 baseSetup({...

REQ-1477: import { screen } from "\_\_support\_\_/ui";  
  
import type { SetupOpts } from "./setup";  
import { setup as baseSetup } from "./setup";  
  
function setup(opts: SetupOpts) {  
 baseSetup({  
 ...

REQ-1478: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1479: import \_ from "underscore";  
  
import { isImplicitDeleteAction } from "metabase/actions/utils";  
import { isNotNull } from "metabase/lib/types";  
import { isEmpty } from "metabase/lib/validate";  
import...

REQ-1480: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1481: import { t } from "ttag";  
  
import type { FieldType, InputSettingType } from "metabase-types/api";  
  
interface FieldOptionType {  
 value: FieldType;  
 name: string;  
}  
  
interface InputOptionType {  
 ...

REQ-1482: import { createContext, useContext } from "react";  
import \_ from "underscore";  
  
import type { ActionFormSettings, WritebackAction } from "metabase-types/api";  
  
import { getDefaultFormSettings } from...

REQ-1483: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1484: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1485: import type Question from "metabase-lib/v1/Question";  
import type NativeQuery from "metabase-lib/v1/queries/NativeQuery";  
import type {  
 ActionFormSettings,  
 FieldType,  
 InputSettingType,  
 ...

REQ-1486: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import {  
 getDefaultFieldSettings,  
 getDefaultFormSettings,  
} from "metabase/actions/utils";  
import Question from...

REQ-1487: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1488: import type { ReactNode } from "react";  
  
import type { WritebackAction } from "metabase-types/api";  
  
export type EditableActionParams = Pick<  
 Partial<WritebackAction>,  
 "name" |...

REQ-1489: import type { WritebackQueryAction } from "metabase-types/api";  
  
import { getDefaultFormSettings } from "./../../../utils";  
  
export function createEmptyWritebackAction():...

REQ-1490: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1491: export { Description } from "./Description";

REQ-1492: import { screen } from "\_\_support\_\_/ui";  
  
import { setup } from "./setup";  
  
describe("FormCerator > Description (OSS)", () => {  
 it("should show a help link when `show-metabase-links: true`", () =>...

REQ-1493: import { screen } from "\_\_support\_\_/ui";  
  
import type { SetupOpts } from "./setup";  
import { setup as baseSetup } from "./setup";  
  
function setup(opts: SetupOpts) {  
 return baseSetup({  
 ...

REQ-1494: import { screen } from "\_\_support\_\_/ui";  
  
import type { SetupOpts } from "./setup";  
import { setup as baseSetup } from "./setup";  
  
function setup(opts: SetupOpts) {  
 return baseSetup({  
 ...

REQ-1495: export { EmptyFormPlaceholder } from "./EmptyFormPlaceholder";

REQ-1496: import { screen } from "\_\_support\_\_/ui";  
  
import { setup } from "./setup";  
  
describe("EmptyFormPlaceholder (OSS)", () => {  
 it("should render help link when `show-metabase-links: true`", () => {  
 ...

REQ-1497: import { screen } from "\_\_support\_\_/ui";  
  
import type { SetupOpts } from "./setup";  
import { setup as baseSetup } from "./setup";  
  
function setup(opts: SetupOpts) {  
 baseSetup({  
 ...

REQ-1498: import { screen } from "\_\_support\_\_/ui";  
  
import type { SetupOpts } from "./setup";  
import { setup as baseSetup } from "./setup";  
  
function setup(opts: SetupOpts) {  
 baseSetup({  
 ...

REQ-1499: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1500: export { FormCreator } from "./FormCreator";  
export \* from "./utils";

REQ-1501: import \_ from "underscore";  
  
import { moveElement } from "metabase/lib/arrays";  
import type { FieldSettingsMap, InputSettingType } from "metabase-types/api";  
  
const inputTypeMap:...

REQ-1502: import { getDefaultFieldSettings } from "metabase/actions/utils";  
import type { FieldSettingsMap } from "metabase-types/api";  
  
import { reorderFields } from "./utils";  
  
describe("actions > containers...

REQ-1503: import { queryIcon, screen } from "\_\_support\_\_/ui";  
import {  
 createMockActionParameter,  
 createMockImplicitQueryAction,  
} from "metabase-types/api/mocks";  
  
import type { SetupOpts } from...

REQ-1504: import userEvent from "@testing-library/user-event";  
  
import { getIcon, queryIcon, screen, waitFor, within } from "\_\_support\_\_/ui";  
import {  
 createMockActionParameter,  
 createMockCard,  
 ...

REQ-1505: export type SideView = "dataReference" | "actionForm" | "actionSettings";  
  
export interface ActionCreatorUIProps {  
 canRename: boolean;  
 canChangeFieldSettings: boolean;  
}

REQ-1506: import \_ from "underscore";  
  
import { getDefaultFieldSettings } from "metabase/actions/utils";  
import type { ActionFormSettings, Parameter } from "metabase-types/api";  
  
export const...

REQ-1507: import {  
 createMockActionFormSettings,  
 createMockFieldSettings,  
 createMockParameter,  
} from "metabase-types/api/mocks";  
  
import { syncFieldsWithParameters } from...

REQ-1508: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1509: export { ActionExecuteModal } from "./ActionExecuteModal";

REQ-1510: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1511: export { ActionPicker, ConnectedActionPicker } from "./ActionPicker";

REQ-1512: import type { CardId, WritebackAction } from "metabase-types/api";  
  
type ModelId = CardId;  
  
export type ModelActionMap = Record<ModelId, WritebackAction[]>;

REQ-1513: import \_ from "underscore";  
  
import type { WritebackAction } from "metabase-types/api";  
  
import type { ModelActionMap } from "./types";  
  
export const sortAndGroupActions = (  
 actions?:...

REQ-1514: import { createMockQueryAction } from "metabase-types/api/mocks";  
  
import { sortAndGroupActions } from "./utils";  
  
const testActions = [  
 createMockQueryAction({ id: 2, name: "Bear Action",...

REQ-1515: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1516: import { useCallback, useMemo } from "react";  
import \_ from "underscore";  
  
import { getForm, getFormValidationSchema } from "metabase/actions/utils";  
import type {  
 ActionFormInitialValues,  
 ...

REQ-1517: import { renderHook } from "@testing-library/react";  
  
import {  
 createMockActionParameter,  
 createMockFieldSettings,  
 createMockQueryAction,  
} from "metabase-types/api/mocks";  
  
import...

REQ-1518: import dayjs from "dayjs";  
  
import type { FieldSettings as LocalFieldSettings } from "metabase/actions/types";  
import { getDefaultFieldSettings } from "metabase/actions/utils";  
import { isEmpty }...

REQ-1519: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import { checkNotNull } from "metabase/lib/types";  
import type { FieldSettingsMap } from "metabase-types/api";  
import {  
 createMockField,  
 ...

REQ-1520: import { useEffect, useMemo } from "react";  
import { useAsyncFn } from "react-use";  
  
import type { ParametersForActionExecution } from "metabase-types/api";  
  
const NO\_VALUES:...

REQ-1521: import type Field from "metabase-lib/v1/metadata/Field";  
import type {  
 ActionFormOption,  
 FieldSettings as BaseFieldSettings,  
 InputComponentType,  
} from "metabase-types/api";  
  
export type...

REQ-1522: import { t } from "ttag";  
import \* as Yup from "yup";  
  
import \* as Errors from "metabase/lib/errors";  
import type Field from "metabase-lib/v1/metadata/Field";  
import { TYPE } from...

REQ-1523: import {  
 getDefaultFieldSettings,  
 getDefaultFormSettings,  
 sortActionParams,  
} from "./utils";  
  
const createParameter = (options?: any) => {  
 return {  
 id: "test\_parameter",  
 name: "Test...

REQ-1524: Reducers needed for admin section (only used in "main" app)

REQ-1525: import { updateSetting } from "metabase/admin/settings/settings";  
import { createAsyncThunk } from "metabase/lib/redux";  
  
import { getCurrentVersion } from "./selectors";  
  
export const disableNotice...

REQ-1526: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1527: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1528: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1529: import { createReducer } from "@reduxjs/toolkit";  
import { t } from "ttag";  
  
import { combineReducers } from "metabase/lib/redux";  
import { isNotNull } from "metabase/lib/types";  
import {  
 ...

REQ-1530: import { getEngines } from "metabase/databases/selectors";  
import { isDeprecatedEngine } from "metabase/lib/engine";  
import { getSetting } from "metabase/selectors/settings";  
import type Database...

REQ-1531: export \* from "./AdminNav";

REQ-1532: export \* from "./SettingsSection";

REQ-1533: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1534: export \* from "./DatabaseList";

REQ-1535: export \* from "./ModelActionsSection";

REQ-1536: export \* from "./ModelCachingControl";

REQ-1537: import { createAction, createReducer } from "@reduxjs/toolkit";  
import { push } from "react-router-redux";  
  
import Databases from "metabase/entities/databases";  
import { combineReducers } from...

REQ-1538: import \_ from "underscore";  
  
import type { DatabaseData } from "metabase-types/api";  
  
export const editParamsForUserControlledScheduling = \_.compose(  
 editScheduleParamsForUserControlledScheduling,  
...

REQ-1539: import { createMockDatabase } from "metabase-types/api/mocks";  
  
import { editParamsForUserControlledScheduling } from "./editParamsForUserControlledScheduling";  
  
it("adds full\_sync param if user will...

REQ-1540: import type { State } from "metabase-types/store";  
  
export const getDeletes = (state: State) => state.admin.databases.deletes;  
  
export const getDeletionError = (state: State) =>  
 ...

REQ-1541: import type { Database, DatabaseFeature, DatabaseId } from "metabase-types/api";  
  
export const isDbModifiable = (  
 database: { id?: DatabaseId; is\_attached\_dwh?: boolean } | undefined,  
) => {  
 ...

REQ-1542: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1543: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1544: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1545: export \* from "./QueryDefinition";

REQ-1546: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1547: import {  
 combineReducers,  
 createAction,  
 createThunkAction,  
 handleActions,  
} from "metabase/lib/redux";  
import { MetabaseApi, RevisionsApi } from "metabase/services";  
  
export const...

REQ-1548: export const getPreviewSummary = (state) =>  
 state.admin.datamodel.previewSummary;  
export const getRevisions = (state) => state.admin.datamodel.revisions;  
export const getCurrentUser = (state) =>...

REQ-1549: import \* as Lib from "metabase-lib";  
import type Metadata from "metabase-lib/v1/metadata/Metadata";  
import type { StructuredQuery, TableId } from "metabase-types/api";  
  
export function...

REQ-1550: import { color } from "metabase/lib/colors";  
import type { GroupId, User } from "metabase-types/api";  
  
export const userToColor = (user: User) => {  
 return user.is\_superuser ? color("accent2") :...

REQ-1551: export \* from "./GroupMembersTable";

REQ-1552: export \* from "./MembershipSelect";

REQ-1553: export type UserStatus = "active" | "deactivated";  
  
export const USER\_STATUS: Record<string, UserStatus> = {  
 active: "active",  
 deactivated: "deactivated",  
};  
  
export const...

REQ-1554: export \* from "./EditUserModal";

REQ-1555: export const CLEAR\_MEMBERSHIPS = "metabase/admin/people/CLEAR\_MEMBERSHIPS";  
export const STORE\_TEMPORARY\_PASSWORD =  
 "metabase/admin/people/STORE\_TEMPORARY\_PASSWORD";  
export const...

REQ-1556: import { useEffect, useMemo, useState } from "react";  
  
import { usePagination } from "metabase/common/hooks/use-pagination";  
import { SEARCH\_DEBOUNCE\_DURATION } from "metabase/lib/constants";  
  
import...

REQ-1557: import {  
 combineReducers,  
 createAction,  
 handleActions,  
} from "metabase/lib/redux";  
  
import { CLEAR\_TEMPORARY\_PASSWORD, STORE\_TEMPORARY\_PASSWORD } from "./events";  
  
// ACTION CREATORS  
  
export...

REQ-1558: import { createSelector } from "@reduxjs/toolkit";  
  
import { ACTIVE\_USERS\_NUDGE\_THRESHOLD } from "metabase/admin/people/constants";  
import { hasAnySsoFeature } from...

REQ-1559: import { t } from "ttag";  
import \* as Yup from "yup";  
  
import type { CacheableModel } from "metabase-types/api";  
import type { AdminPath } from "metabase-types/store";  
  
import { PerformanceTabId,...

REQ-1560: export const rootId = 0;  
export const defaultMinDurationMs = 1000;  
export const defaultCronSchedule = "0 0 \* \* \* ? \*";

REQ-1561: import type { AnySchema } from "yup";  
  
import type { CacheableModel } from "metabase-types/api";  
  
export type UpdateTargetId = (  
 newTargetId: number | null,  
 isFormDirty: boolean,  
) =>...

REQ-1562: import type {  
 AdaptiveStrategy,  
 InheritStrategy,  
 ScheduleSettings,  
} from "metabase-types/api";  
  
import {  
 cronToScheduleSettings,  
 getShortStrategyLabel,  
 hourTo24HourFormat,  
 ...

REQ-1563: export \* from "./ApplicationPermissionsHelp";

REQ-1564: export \* from "./CollectionPermissionsHelp";

REQ-1565: export \* from "./DataPermissionsHelp";

REQ-1566: export \* from "./EntityViewSwitch";

REQ-1567: export \* from "./FilterableTree";

REQ-1568: export const searchItems = (items: any[], filter: string) => {  
 const matchingItems = items.filter((item) =>  
 item.name.toLowerCase().includes(filter),  
 );  
  
 const children = items  
 .map((c)...

REQ-1569: import { searchItems } from "./utils";  
  
const items = [  
 {  
 name: "Parent node",  
 children: [  
 { name: "child", children: [{ name: "Lowest node" }] },  
 { name: "aLsO cHiLd" },  
 ...

REQ-1570: export \* from "./PermissionHelpDescription";

REQ-1571: export \* from "./PermissionsEditor";  
export \* from "./PermissionsEditorEmptyState";

REQ-1572: export \* from "./PermissionsPageLayout";

REQ-1573: import userEvent from "@testing-library/user-event";  
  
import { fireEvent, getIcon, render, screen } from "\_\_support\_\_/ui";  
  
import { DataPermissionValue } from "../../types";  
  
import {...

REQ-1574: export \* from "./PermissionsSelect";

REQ-1575: export \* from "./PermissionsSidebar";

REQ-1576: export \* from "./PermissionsTable";

REQ-1577: export \* from "./ToolbarButton";

REQ-1578: eslint-disable ttag/no-module-declaration -- see metabase#55045

REQ-1579: import { t } from "ttag";  
  
// eslint-disable-next-line ttag/no-module-declaration -- see metabase#55045  
export const UNABLE\_TO\_CHANGE\_ADMIN\_PERMISSIONS = t`Administrators always have the highest...

REQ-1580: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1581: import { assocIn, merge } from "icepick";  
import { push } from "react-router-redux";  
import { t } from "ttag";  
  
import {  
 inferAndUpdateEntityPermissions,  
 ...

REQ-1582: import { createSelector } from "@reduxjs/toolkit";  
import { getIn } from "icepick";  
import { t } from "ttag";  
import \_ from "underscore";  
  
import {  
 isInstanceAnalyticsCollection,  
 ...

REQ-1583: import { t } from "ttag";  
  
import { isNotFalsy } from "metabase/lib/types";  
import type Metadata from "metabase-lib/v1/metadata/Metadata";  
import type Schema from...

REQ-1584: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import {  
 createMockDatabase,  
 createMockSchema,  
 createMockTable,  
} from "metabase-types/api/mocks";  
  
import {...

REQ-1585: import { createMockSettingsState } from "metabase-types/store/mocks";  
  
import { DataPermission, DataPermissionValue } from "../../types";  
  
// Database 2 contains an imaginary multi-schema database...

REQ-1586: import type { Selector } from "@reduxjs/toolkit";  
import { createSelector } from "@reduxjs/toolkit";  
import { t } from "ttag";  
  
import type { ITreeNodeItem } from...

REQ-1587: import type { State } from "metabase-types/store";  
  
import type { RawDataRouteParams } from "../../types";  
  
import { state as mockState } from "./data-permissions.unit.spec.fixtures";  
  
import {...

REQ-1588: import { createSelector } from "@reduxjs/toolkit";  
import \_ from "underscore";  
  
import { diffDataPermissions } from "metabase/admin/permissions/utils/graph";  
import { PLUGIN\_DATA\_PERMISSIONS } from...

REQ-1589: import \_ from "underscore";  
  
import { getNativePermissionDisabledTooltip } from "metabase/admin/permissions/selectors/data-permissions/shared";  
import {  
 getFieldsPermission,  
 ...

REQ-1590: import { createSelector } from "@reduxjs/toolkit";  
import { t } from "ttag";  
  
import { getGroupNameLocalized } from "metabase/lib/groups";  
import type { Group } from "metabase-types/api";  
import type...

REQ-1591: import { assocIn } from "icepick";  
  
import { PLUGIN\_ADVANCED\_PERMISSIONS } from "metabase/plugins";  
import type { State } from "metabase-types/store";  
  
import { DataPermission, DataPermissionValue }...

REQ-1592: import { createSelector } from "@reduxjs/toolkit";  
import \_ from "underscore";  
  
import Groups from "metabase/entities/groups";  
import { isAdminGroup, isDefaultGroup } from...

REQ-1593: export \* from "./breadcrumbs";  
export \* from "./data-sidebar";  
export \* from "./diff";  
export \* from "./fields";  
export \* from "./group-sidebar";  
export \* from "./permission-editor";  
export \* from...

REQ-1594: import type { Selector } from "@reduxjs/toolkit";  
import { createSelector } from "@reduxjs/toolkit";  
import { msgid, ngettext, t } from "ttag";  
import \_ from "underscore";  
  
import Groups from...

REQ-1595: import { createSelector } from "@reduxjs/toolkit";  
  
import type { State } from "metabase-types/store";  
  
import { getIsDirty } from "./diff";  
  
export const showRevisionChangedModal = createSelector(  
 ...

REQ-1596: import \_ from "underscore";  
  
import { getNativePermissionDisabledTooltip } from "metabase/admin/permissions/selectors/data-permissions/shared";  
import { getSchemasPermission } from...

REQ-1597: import {  
 NATIVE\_PERMISSION\_REQUIRES\_DATA\_ACCESS,  
 UNABLE\_TO\_CHANGE\_ADMIN\_PERMISSIONS,  
 UNABLE\_TO\_CHANGE\_LEGACY\_PERMISSIONS,  
} from "metabase/admin/permissions/constants/messages";  
import {...

REQ-1598: import \_ from "underscore";  
  
import { getNativePermissionDisabledTooltip } from "metabase/admin/permissions/selectors/data-permissions/shared";  
import {  
 getSchemasPermission,  
 ...

REQ-1599: import type { State } from "metabase-types/store";  
  
export const getIsHelpReferenceOpen = (state: State) => {  
 return state.admin.permissions.isHelpReferenceOpen;  
};

REQ-1600: import type { ReactNode } from "react";  
  
export type GroupRouteParams = {  
 groupId?: number;  
 databaseId?: number;  
 schemaName?: string;  
};  
  
export type RawGroupRouteParams = {  
 groupId?:...

REQ-1601: import { checkNotNull } from "metabase/lib/types";  
import type Database from "metabase-lib/v1/metadata/Database";  
import type Schema from "metabase-lib/v1/metadata/Schema";  
import type Table from...

REQ-1602: import { getIn } from "icepick";  
  
import type {  
 DatabaseEntityId,  
 EntityId,  
 SchemaEntityId,  
 TableEntityId,  
} from "metabase/admin/permissions/types";  
import {  
 DataPermission,  
 ...

REQ-1603: import \_ from "underscore";  
  
import type {  
 DataPermission,  
 DataPermissionValue,  
 DatabaseEntityId,  
 EntityWithGroupId,  
 SchemaEntityId,  
} from "metabase/admin/permissions/types";  
import {...

REQ-1604: import \_ from "underscore";  
  
import {  
 DataPermission,  
 DataPermissionValue,  
} from "metabase/admin/permissions/types";  
import { PLUGIN\_ADVANCED\_PERMISSIONS } from "metabase/plugins";  
import type {...

REQ-1605: export \* from "./get";  
export \* from "./has";  
export \* from "./update";  
export \* from "./utils";

REQ-1606: import { getIn, setIn } from "icepick";  
import \_ from "underscore";  
  
import type {  
 DatabaseEntityId,  
 EntityId,  
 SchemaEntityId,  
 TableEntityId,  
} from "metabase/admin/permissions/types";  
import...

REQ-1607: import { DataPermissionValue } from "metabase/admin/permissions/types";  
import { PLUGIN\_ADVANCED\_PERMISSIONS } from "metabase/plugins";  
  
export const isRestrictivePermission = (value:...

REQ-1608: export \* from "./data-permissions";  
export \* from "./permissions-diff";

REQ-1609: import \_ from "underscore";  
  
import type {  
 CollectionPermissions,  
 GroupsPermissions,  
} from "metabase-types/api";  
  
// utils for dealing with partial graph updates  
  
// select only the parts of the...

REQ-1610: import type { GroupsPermissions } from "metabase-types/api";  
  
import { DataPermission, DataPermissionValue } from "../../types";  
  
import {  
 getModifiedCollectionPermissionsGraphParts,  
 ...

REQ-1611: import type Database from "metabase-lib/v1/metadata/Database";  
import type {  
 ConcreteTableId,  
 Group,  
 GroupsPermissions,  
} from "metabase-types/api";  
  
import { DataPermission } from...

REQ-1612: import type Metadata from "metabase-lib/v1/metadata/Metadata";  
import type Table from "metabase-lib/v1/metadata/Table";  
import type { ConcreteTableId } from "metabase-types/api";  
  
import type {...

REQ-1613: import {  
 isDatabaseEntityId,  
 isSchemaEntityId,  
 isTableEntityId,  
} from "./data-entity-id";  
  
export const DATABASES\_BASE\_PATH = `/admin/permissions/data/database`;  
export const GROUPS\_BASE\_PATH...

REQ-1614: import {  
 getDatabaseFocusPermissionsUrl,  
 getGroupFocusPermissionsUrl,  
} from "./urls";  
  
describe("getDatabaseFocusPermissionsUrl", () => {  
 it("when entityId is not specified it returns base...

REQ-1615: import { trackSchemaEvent } from "metabase/lib/analytics";  
  
export const trackTrackingPermissionChanged = (isEnabled: boolean) => {  
 trackSchemaEvent("settings", {  
 event: isEnabled  
 ?...

REQ-1616: export \* from "./AuthCard";

REQ-1617: export \* from "./GoogleAuthForm";

REQ-1618: import \* as Yup from "yup";  
  
import \* as Errors from "metabase/lib/errors";  
import type { SettingDefinition } from "metabase-types/api";  
  
const REQUIRED\_SCHEMA = {  
 is: (isEnabled: boolean,...

REQ-1619: export \* from "./GoogleAuthCard";

REQ-1620: export \* from "./LdapAuthCard";

REQ-1621: import { t } from "ttag";  
import \* as Yup from "yup";  
  
import type { ApiKey } from "metabase-types/api";  
  
export function formatMaskedKey(maskedKey: string) {  
 return maskedKey.substring(0, 7) +...

REQ-1622: export \* from "./CloudPanel";

REQ-1623: import dayjs from "dayjs";  
  
import type {  
 CloudMigration,  
 CloudMigrationState,  
} from "metabase-types/api/cloud-migration";  
  
export type InternalCloudMigrationState = CloudMigrationState |...

REQ-1624: export \* from "./SMTPConnectionCard";

REQ-1625: import { trackSimpleEvent } from "metabase/lib/analytics";  
import type {  
 CustomSMTPSetupClickedEvent,  
 CustomSMTPSetupSuccessEvent,  
} from "metabase-types/analytics";  
  
export function...

REQ-1626: export \* from "./EmbeddingLegaleseModal";

REQ-1627: export \* from "./EmbeddingSdkOptionCard";

REQ-1628: export \* from "./InteractiveEmbeddingOptionCard";

REQ-1629: export \* from "./LinkButton";

REQ-1630: export \* from "./StaticEmbeddingOptionCard";

REQ-1631: export { StaticEmbeddingOptionCard } from "./StaticEmbeddingOptionCard";  
export { EmbeddingSdkOptionCard } from "./EmbeddingSdkOptionCard";  
export { InteractiveEmbeddingOptionCard } from...

REQ-1632: export type EmbeddingOptionIconProps = {  
 disabled?: boolean;  
};

REQ-1633: export \* from "./EmbeddingSdkSettings";

REQ-1634: export \* from "./EmbeddingToggle";

REQ-1635: export { StaticEmbeddingSettings } from "./StaticEmbeddingSettings";  
export { EmbeddingSdkSettings } from "./EmbeddingSdkSettings/EmbeddingSdkSettings";

REQ-1636: export \* from "./LicenseInput";

REQ-1637: export \* from "./SettingHeader";

REQ-1638: export \* from "./SettingsLicense";

REQ-1639: export \* from "./SettingsNav";

REQ-1640: import type { Database, SchemaName } from "metabase-types/api";  
  
export const getDatabaseOptions = (databases: Database[]) =>  
 databases.map((db) => ({  
 label: db.router\_user\_attribute  
 ?...

REQ-1641: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import { checkNotNull } from "metabase/lib/types";  
import type { Database, Schema } from "metabase-types/api";  
import { createMockDatabase,...

REQ-1642: export \* from "./GroupMappingsWidget";

REQ-1643: import type { NotificationChannel } from "metabase-types/api";  
  
import type { WebhookFormProps } from "./WebhookForm";  
  
export const buildAuthInfo = (  
 form: WebhookFormProps,  
): Record<string,...

REQ-1644: import {  
 createMockChannel,  
 createMockChannelDetails,  
} from "metabase-types/api/mocks/channel";  
  
import type { WebhookFormProps } from "./WebhookForm";  
import { buildAuthInfo, channelToForm }...

REQ-1645: export \* from "./PublicLinksListing";  
export \* from "./EmbeddedResources";  
export \* from "./PublicResources";

REQ-1646: export { VersionUpdateNotice } from "./VersionUpdateNotice";

REQ-1647: import {  
 combineReducers,  
 createAction,  
 createThunkAction,  
 handleActions,  
} from "metabase/lib/redux";  
import { refreshSiteSettings } from "metabase/redux/settings";  
import { SettingsApi }...

REQ-1648: import type { ComponentType } from "react";  
  
import type {  
 SettingDefinition,  
 SettingKey,  
 SettingValue,  
 Settings,  
} from "metabase-types/api";  
  
export type SettingElement<Key extends...

REQ-1649: import { t } from "ttag";  
  
import { useDocsUrl } from "metabase/common/hooks";  
  
export const settingToFormField = (setting) => ({  
 name: setting.key,  
 label: setting.display\_name,  
 description:...

REQ-1650: export { Help } from "./Help";

REQ-1651: export \* from "./LogLevelsModal";

REQ-1652: import type { LoggerDurationUnit } from "metabase-types/api";  
  
// Some options are not practically useful  
export type AllowedTimeUnit = Exclude<  
 LoggerDurationUnit,  
 "nanoseconds" | "microseconds"...

REQ-1653: import type { LoggerPreset } from "metabase-types/api";  
  
export function getPresetJson(preset: LoggerPreset) {  
 const logLevels = Object.fromEntries(  
 preset.loggers.map(({ level, name }) =>...

REQ-1654: import { useInterval } from "@mantine/hooks";  
import { useCallback, useEffect, useRef, useState } from "react";  
import { useMount, useUnmount } from "react-use";  
import { t } from "ttag";  
  
import {...

REQ-1655: export \* from "./Logs";

REQ-1656: import dayjs from "dayjs";  
import orderBy from "lodash.orderby";  
import \_ from "underscore";  
  
import {  
 type QueryParam,  
 type UrlStateConfig,  
 getFirstParamValue,  
} from...

REQ-1657: export \* from "./ModelCacheRefreshJobs";  
export \* from "./ModelCacheRefreshJobModal";

REQ-1658: export \* from "./TaskModal";

REQ-1659: export \* from "./TaskPicker";

REQ-1660: export \* from "./TaskStatusPicker";

REQ-1661: export \* from "./TasksApp";

REQ-1662: import {  
 type QueryParam,  
 type UrlStateConfig,  
 getFirstParamValue,  
} from "metabase/common/hooks/use-url-state";  
import type { ListTasksSortColumn, TaskStatus } from...

REQ-1663: export \* from "./ToolsUpsell";

REQ-1664: export type MappingsType = Record<string, number[]>;  
export type GroupIds = number[];  
export type DeleteMappingModalValueType = "nothing" | "clear" | "delete";  
export type UserGroupType = { id:...

REQ-1665: import { trackSchemaEvent } from "metabase/lib/analytics";  
  
type UpsellEventProps = {  
 location: string;  
 campaign: string;  
};  
  
export const trackUpsellViewed = ({ location, campaign }:...

REQ-1666: export \* from "./UpsellBanner";  
export \* from "./UpsellBigCard";  
export \* from "./UpsellCard";  
export \* from "./UpsellPill";  
export { UpsellGem } from "./UpsellGem";

REQ-1667: import { useUrlWithUtm } from "metabase/common/hooks";  
  
interface UpsellLinkProps {  
 /\* The URL we're sending them to \*/  
 url: string | undefined;  
 /\* The name of the feature we're trying to sell...

REQ-1668: export const UPGRADE\_URL = "https://www.metabase.com/upgrade";

REQ-1669: export \* from "./UpsellBetterSupport";  
export \* from "./UpsellCacheConfig";  
export \* from "./UpsellCloud";  
export \* from "./UpsellEmailWhitelabel";  
export \* from "./UpsellHosting";  
export \* from...

REQ-1670: Upsells Components

REQ-1671: import { push, replace, routerActions } from "react-router-redux";  
import { connectedReduxRedirect } from "redux-auth-wrapper/history3/redirect";  
  
import { getAdminPaths } from...

REQ-1672: import \_ from "underscore";  
  
import type {  
 CreateActionRequest,  
 GetActionRequest,  
 ListActionsRequest,  
 UpdateActionRequest,  
 WritebackAction,  
 WritebackActionId,  
} from...

REQ-1673: import type {  
 CreateRecentRequest,  
 Field,  
 PopularItem,  
 PopularItemsResponse,  
 RecentItem,  
 RecentsRequest,  
 RecentsResponse,  
 VisualizationDisplay,  
} from "metabase-types/api";  
  
import {...

REQ-1674: import type {  
 ApiKey,  
 ApiKeyId,  
 CreateApiKeyRequest,  
 CreateApiKeyResponse,  
 RegenerateApiKeyResponse,  
 UpdateApiKeyRequest,  
 UpdateApiKeyResponse,  
} from...

REQ-1675: import {  
 buildCreateApi,  
 coreModule,  
 reactHooksModule,  
 skipToken,  
} from "@reduxjs/toolkit/query/react";  
import {  
 createDispatchHook,  
 createSelectorHook,  
 createStoreHook,  
} from...

REQ-1676: import type {  
 Dashboard,  
 DashboardQueryMetadata,  
 DatabaseId,  
 DatabaseXray,  
 GetXrayDashboardQueryMetadataRequest,  
} from "metabase-types/api";  
  
import { Api } from "./api";  
import {  
 ...

REQ-1677: import type {  
 Bookmark,  
 CreateBookmarkRequest,  
 DeleteBookmarkRequest,  
 ReorderBookmarksRequest,  
} from "metabase-types/api";  
  
import { Api } from "./api";  
import {  
 idTag,  
 invalidateTags,  
 ...

REQ-1678: import type { ErrorPayload } from "metabase-types/api";  
  
import { Api } from "./api";  
  
interface BugReportResponse {  
 success: boolean;  
}  
  
export const bugReportApi = Api.injectEndpoints({  
 ...

REQ-1679: import { PLUGIN\_API } from "metabase/plugins";  
import type {  
 Card,  
 CardId,  
 CardQueryMetadata,  
 CardQueryRequest,  
 CollectionItem,  
 CreateCardFromCsvRequest,  
 CreateCardRequest,  
 ...

REQ-1680: import type { ChannelDetails, NotificationChannel } from "metabase-types/api";  
  
import { Api } from "./api";  
import { idTag, invalidateTags, listTag, provideChannelListTags } from "./tags";  
  
const...

REQ-1681: import type { CloudMigration } from "metabase-types/api/cloud-migration";  
  
import { Api } from "./api";  
import { listTag } from "./tags";  
  
export const clouldMigrationApi = Api.injectEndpoints({  
 ...

REQ-1682: import type {  
 Collection,  
 CreateCollectionRequest,  
 DeleteCollectionRequest,  
 GetCollectionDashboardQuestionCandidatesRequest,  
 GetCollectionDashboardQuestionCandidatesResult,  
 ...

REQ-1683: import { PLUGIN\_API } from "metabase/plugins";  
import type {  
 CopyDashboardRequest,  
 CreateDashboardRequest,  
 Dashboard,  
 DashboardId,  
 DashboardQueryMetadata,  
 FieldId,  
 FieldValue,  
 ...

REQ-1684: import type {  
 AutocompleteRequest,  
 AutocompleteSuggestion,  
 CardAutocompleteRequest,  
 CardAutocompleteSuggestion,  
 CreateDatabaseRequest,  
 Database,  
 DatabaseId,  
 Field,  
 ...

REQ-1685: import type {  
 CardQueryMetadata,  
 Dataset,  
 DatasetQuery,  
 FieldValue,  
 GetRemappedParameterValueRequest,  
 NativeDatasetResponse,  
} from "metabase-types/api";  
  
import { Api } from...

REQ-1686: import type { EmailSMTPSettings } from "metabase-types/api";  
  
import { Api } from "./api";  
import { invalidateTags, tag } from "./tags";  
  
export const settingsApi = Api.injectEndpoints({  
 endpoints:...

REQ-1687: import {  
 type BaseEntityId,  
 isBaseEntityID,  
} from "metabase-types/api/entity-id";  
  
import { Api } from "./api";  
  
const validEntityTypes = [  
 "action",  
 "card",  
 "collection",  
 "dashboard",  
 ...

REQ-1688: import type {  
 CreateFieldDimensionRequest,  
 Field,  
 FieldDimension,  
 FieldId,  
 FieldValue,  
 GetFieldRequest,  
 GetFieldValuesResponse,  
 GetRemappedFieldValueRequest,  
 ...

REQ-1689: import type { Feature, FeatureCollection } from "geojson";  
import { t } from "ttag";  
  
import { computeMinimalBounds } from "metabase/visualizations/lib/mapping";  
import type { GeoJSONData } from...

REQ-1690: import type { EnterpriseSettings } from "metabase-types/api";  
  
import { Api } from "./api";  
import { invalidateTags, tag } from "./tags";  
  
type GoogleAuthSettings = Pick<  
 EnterpriseSettings,  
 |...

REQ-1691: export \* from "./action";  
export \* from "./activity";  
export \* from "./api";  
export \* from "./api-key";  
export \* from "./automagic-dashboards";  
export \* from "./bookmark";  
export \* from...

REQ-1692: import type { EnterpriseSettings } from "metabase-types/api";  
  
import { Api } from "./api";  
import { invalidateTags, tag } from "./tags";  
  
type LdapSettings = Pick<  
 EnterpriseSettings,  
 |...

REQ-1693: import type { AdjustLogLevelsRequest, LoggerPreset } from "metabase-types/api";  
  
import { Api } from "./api";  
import { provideLoggerPresetListTags } from "./tags";  
  
export const loggerApi =...

REQ-1694: import type { UserLoginHistory } from "metabase-types/api";  
  
import { Api } from "./api";  
  
export const loginHistoryApi = Api.injectEndpoints({  
 endpoints: (builder) => ({  
 getLoginHistory:...

REQ-1695: import type {  
 ModelIndex,  
 ModelIndexCreateQuery,  
 ModelIndexDeleteQuery,  
 ModelIndexesListQuery,  
} from "metabase-types/api";  
  
import { Api } from "./api";  
import { invalidateTags, listTag,...

REQ-1696: import type { VerifyItemRequest } from "metabase-types/api";  
  
import { Api } from "./api";  
import { invalidateTags, provideModeratedItemTags } from "./tags";  
  
export const contentVerificationApi =...

REQ-1697: import type {  
 CreateNotificationRequest,  
 ListNotificationsRequest,  
 Notification,  
 NotificationId,  
 UpdateNotificationRequest,  
} from "metabase-types/api/notification";  
  
import { Api } from...

REQ-1698: import type {  
 GetParameterValuesRequest,  
 ParameterValues,  
 SearchParameterValuesRequest,  
} from "metabase-types/api";  
  
import { Api } from "./api";  
import { idTag } from "./tags";  
  
export const...

REQ-1699: import type {  
 BaseGroupInfo,  
 CreateMembershipRequest,  
 Group,  
 GroupId,  
 GroupListQuery,  
 ListUserMembershipsResponse,  
 Membership,  
} from "metabase-types/api";  
  
import { Api } from...

REQ-1700: import type {  
 CardId,  
 ListPersistedInfoRequest,  
 ListPersistedInfoResponse,  
 ModelCacheRefreshStatus,  
 PersistedInfoId,  
 PersistedInfoRefreshSchedule,  
} from "metabase-types/api";  
  
import {...

REQ-1701: import { Api } from "./api";  
  
const productFeedbackApi = Api.injectEndpoints({  
 endpoints: (builder) => ({  
 sendProductFeedback: builder.mutation<  
 void,  
 { comment?: string; email?:...

REQ-1702: import type { BaseQueryFn } from "@reduxjs/toolkit/query/react";  
  
import api from "metabase/lib/api";  
  
type AllowedHTTPMethods = "GET" | "POST" | "PUT" | "DELETE";  
const allowedHTTPMethods = new...

REQ-1703: import type {  
 ListRevisionRequest,  
 RevertRevisionRequest,  
 Revision,  
} from "metabase-types/api";  
  
import { Api } from "./api";  
import { invalidateTags, listTag, provideRevisionListTags } from...

REQ-1704: import { trackSearchRequest } from "metabase/search/analytics";  
import type { SearchRequest, SearchResponse } from "metabase-types/api";  
  
import { Api } from "./api";  
import {...

REQ-1705: import type {  
 CreateSegmentRequest,  
 DeleteSegmentRequest,  
 Segment,  
 SegmentId,  
 UpdateSegmentRequest,  
} from "metabase-types/api";  
  
import { Api } from "./api";  
import {  
 idTag,  
 ...

REQ-1706: import MetabaseSettings from "metabase/lib/settings";  
import { loadSettings } from "metabase/redux/settings";  
import type {  
 EnterpriseSettings,  
 PasswordResetTokenStatus,  
} from...

REQ-1707: import \_ from "underscore";  
  
import type {  
 EnterpriseSettingKey,  
 EnterpriseSettingValue,  
 EnterpriseSettings,  
 SettingDefinition,  
 SettingDefinitionMap,  
} from "metabase-types/api";  
  
import {...

REQ-1708: import type { EnterpriseSettings } from "metabase-types/api";  
  
import { Api } from "./api";  
  
type SlackSettings = Pick<  
 EnterpriseSettings,  
 "slack-app-token" | "slack-bug-report-channel" |...

REQ-1709: import type {  
 CreateSnippetRequest,  
 ListSnippetsParams,  
 NativeQuerySnippet,  
 NativeQuerySnippetId,  
 UpdateSnippetRequest,  
} from "metabase-types/api";  
  
import { Api } from "./api";  
import {  
 ...

REQ-1710: import type {  
 ChannelApiResponse,  
 CreateSubscriptionRequest,  
 DashboardSubscription,  
 ListSubscriptionsRequest,  
 UpdateSubscriptionRequest,  
} from "metabase-types/api";  
  
import { Api } from...

REQ-1711: import type {  
 Field,  
 GetTableQueryMetadataRequest,  
 GetTableRequest,  
 Table,  
 TableId,  
 TableListQuery,  
 UpdateTableFieldsOrderRequest,  
 UpdateTableListRequest,  
 UpdateTableRequest,  
} from...

REQ-1712: export type TagType = (typeof TAG\_TYPES)[number];  
  
export const TAG\_TYPES = [  
 "action",  
 "alert",  
 "api-key",  
 "bookmark",  
 "card",  
 "channel",  
 "cloud-migration",  
 "collection",  
 ...

REQ-1713: export \* from "./constants";  
export \* from "./utils";

REQ-1714: import type { TagDescription } from "@reduxjs/toolkit/query";  
  
import { isVirtualDashCard } from "metabase/dashboard/utils";  
import type {  
 Alert,  
 ApiKey,  
 Bookmark,  
 Card,  
 CardId,  
 ...

REQ-1715: import type {  
 ListTasksRequest,  
 ListTasksResponse,  
 Task,  
 TaskInfo,  
} from "metabase-types/api";  
  
import { Api } from "./api";  
import {  
 provideTaskListTags,  
 provideTaskTags,  
 ...

REQ-1716: import type {  
 CreateTimelineEventRequest,  
 TimelineEvent,  
 TimelineEventId,  
 UpdateTimelineEventRequest,  
} from "metabase-types/api";  
  
import { Api } from "./api";  
import {  
 idTag,  
 ...

REQ-1717: import type {  
 CreateTimelineRequest,  
 GetTimelineRequest,  
 ListCollectionTimelinesRequest,  
 ListTimelinesRequest,  
 Timeline,  
 TimelineId,  
 UpdateTimelineRequest,  
} from...

REQ-1718: import type {  
 DeleteUserKeyValueRequest,  
 GetUserKeyValueRequest,  
 UpdateUserKeyValueRequest,  
} from "metabase-types/api";  
  
import { Api } from "./api";  
  
export const userKeyValueApi =...

REQ-1719: import { STORE\_TEMPORARY\_PASSWORD } from "metabase/admin/people/events";  
import { userUpdated } from "metabase/redux/user";  
import type {  
 CreateUserRequest,  
 ListUsersRequest,  
 ...

REQ-1720: import { t } from "ttag";  
  
type ErrorPayload =  
 | { message: string }  
 | { error: string }  
 | { error\_message: string }  
 | string;  
  
export const getErrorMessage = (  
 payload:  
 | unknown  
 |...

REQ-1721: import { getErrorMessage } from "./errors";  
  
describe("getErrorMessage", () => {  
 it("should return a message from a string payload", () => {  
 const result = getErrorMessage("Some error...

REQ-1722: export \* from "./errors";  
export \* from "./settings";  
export \* from "./use-token-refresh";

REQ-1723: # API Utils

REQ-1724: import { useCallback } from "react";  
import { t } from "ttag";  
  
import { useToast } from "metabase/common/hooks";  
import type {  
 EnterpriseSettingKey,  
 EnterpriseSettingValue,  
 ...

REQ-1725: import { useEffect } from "react";  
  
import { Api, useGetSettingsQuery } from "metabase/api";  
import { useDispatch } from "metabase/lib/redux";  
  
const REFRESH\_INTERVAL = 10 \* 1000; // 10 seconds  
  
/\*\*  
...

REQ-1726: /\*  
 \* This file is subject to the terms and conditions defined in  
 \* file 'LICENSE-EMBEDDING.txt', which is part of this source code package.  
 \*/  
  
import { isWithinIframe } from...

REQ-1727: Enables hot reload in development and noop in production  
MUST be imported BEFORE `react` and `react-dom`

REQ-1728: import { init } from "./app";  
import { publicReducers } from "./reducers-public";  
import { getRoutes } from "./routes-public";  
  
init(publicReducers, getRoutes, () => {});

REQ-1729: import "@mantine/core/styles.css";  
import "@mantine/dates/styles.css";  
  
import "regenerator-runtime/runtime";  
  
// This is conditionally aliased in the webpack config.  
// If EE isn't enabled, it loads...

REQ-1730: import { push } from "react-router-redux";  
import { t } from "ttag";  
  
import { createThunkAction } from "metabase/lib/redux";  
import { addUndo } from "metabase/redux/undo";  
  
export const...

REQ-1731: import { trackSchemaEvent } from "metabase/lib/analytics";  
import type { MoveToTrashEvent } from "metabase-types/analytics";  
  
export const archiveAndTrack = async ({  
 archive,  
 model,  
 modelId,  
 ...

REQ-1732: import { c } from "ttag";  
import \_ from "underscore";  
  
import type { Collection, CollectionItem } from "metabase-types/api";  
  
/\*\*  
 \* @param updateActionResult - result value of await...

REQ-1733: import { type UnknownAction, createAction } from "@reduxjs/toolkit";  
import { getIn } from "icepick";  
import { push } from "react-router-redux";  
  
import { deleteSession, initiateSLO } from...

REQ-1734: export \* from "./AuthButton";

REQ-1735: export \* from "./AuthLayout";

REQ-1736: export \* from "./ForgotPassword";

REQ-1737: export \* from "./ForgotPasswordForm";

REQ-1738: export \* from "./GoogleButton";

REQ-1739: export \* from "./Login";

REQ-1740: export \* from "./LoginForm";

REQ-1741: export \* from "./Logout";

REQ-1742: export \* from "./PasswordButton";

REQ-1743: export \* from "./PasswordPanel";

REQ-1744: export \* from "./ResetPassword";

REQ-1745: export \* from "./ResetPasswordForm";

REQ-1746: import { createSelector } from "@reduxjs/toolkit";  
  
import { PLUGIN\_AUTH\_PROVIDERS } from "metabase/plugins";  
import type { AuthProvider } from "metabase/plugins/types";  
import { getSetting,...

REQ-1747: export interface LoginData {  
 username: string;  
 password: string;  
 remember?: boolean;  
}  
  
export interface ForgotPasswordData {  
 email: string;  
}  
  
export interface ResetPasswordData {  
 ...

REQ-1748: export const RELOAD\_INTERVAL = 2000;

REQ-1749: import fetchMock from "fetch-mock";  
  
import {  
 renderWithProviders,  
 screen,  
 waitFor,  
 waitForLoaderToBeRemoved,  
} from "\_\_support\_\_/ui";  
  
// import { RELOAD\_INTERVAL } from...

REQ-1750: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1751: import { trackSimpleEvent } from "metabase/lib/analytics";  
  
export const trackAddDatabaseDBList = () => {  
 trackSimpleEvent({  
 event: "database\_add\_clicked",  
 triggered\_from: "db-list",  
 ...

REQ-1752: import { trackSimpleEvent } from "metabase/lib/analytics";  
  
export const trackNewMetricInitiated = () =>  
 trackSimpleEvent({  
 event: "plus\_button\_clicked",  
 triggered\_from: "metric",  
 });

REQ-1753: import {  
 createMockRecentCollectionItem,  
 createMockSearchResult,  
} from "metabase-types/api/mocks";  
  
import type { MetricResult, RecentMetric } from "./types";  
  
export const...

REQ-1754: import { t } from "ttag";  
  
import { getCollectionPathAsString } from "metabase/collections/utils";  
import { formatValue } from "metabase/lib/formatting";  
import { isDate } from...

REQ-1755: import {  
 createMockCollection,  
 createMockColumn,  
 createMockDataset,  
 createMockDatasetData,  
} from "metabase-types/api/mocks";  
import { SortDirection } from...

REQ-1756: export { ModelsVideo, ModelsVideoThumbnail } from "./ModelsVideo";

REQ-1757: import { trackSchemaEvent, trackSimpleEvent } from "metabase/lib/analytics";  
import type { CardId } from "metabase-types/api";  
  
export const trackModelClick = (modelId: CardId) =>  
 ...

REQ-1758: import type { RecentCollectionItem } from "metabase-types/api";  
import {  
 createMockRecentCollectionItem,  
 createMockSearchResult,  
} from "metabase-types/api/mocks";  
  
import type { ModelResult,...

REQ-1759: import { t } from "ttag";  
  
import { getCollectionPathAsString } from "metabase/collections/utils";  
import { entityForObject } from "metabase/lib/schema";  
import type { IconName } from...

REQ-1760: import { setupDatabaseEndpoints } from "\_\_support\_\_/server-mocks";  
import { renderWithProviders, screen } from "\_\_support\_\_/ui";  
import { createMockDatabase } from "metabase-types/api/mocks";  
  
import...

REQ-1761: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1762: import { t } from "ttag";  
  
import { skipToken, useGetDatabaseQuery } from "metabase/api";  
import Databases from "metabase/entities/databases";  
import { useSelector } from "metabase/lib/redux";  
import...

REQ-1763: import { trackSchemaEvent } from "metabase/lib/analytics";  
import type { ConcreteTableId } from "metabase-types/api";  
  
export const trackTableClick = (tableId: ConcreteTableId) =>  
 ...

REQ-1764: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1765: import type { CollectionItemModel } from "metabase-types/api";  
  
export const COLLECTION\_PAGE\_SIZE = 25;  
  
export const COLLECTION\_CONTENT\_COLUMNS = [  
 "type",  
 "name",  
 "lastEditedBy",  
 ...

REQ-1766: export \* from "./CollectionContent";  
export \* from "./constants";

REQ-1767: import type { DragEvent, DragEventHandler } from "react";  
import type { DropzoneRootProps } from "react-dropzone";  
  
export const composeFileEventHandler =  
 (fn: DragEventHandler<HTMLElement> |...

REQ-1768: import type { DragEvent } from "react";  
import type { DropzoneRootProps } from "react-dropzone";  
  
import { composeFileEventHandler, getComposedDragProps } from "./utils";  
  
describe("Collections >...

REQ-1769: import { trackSimpleEvent } from "metabase/lib/analytics";  
  
export const trackCollectionNewButtonClicked = () =>  
 trackSimpleEvent({  
 event: "new\_button\_clicked",  
 triggered\_from:...

REQ-1770: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1771: import { trackSimpleEvent } from "metabase/lib/analytics";  
  
export const trackNewCollectionFromHeaderInitiated = () =>  
 trackSimpleEvent({  
 event: "plus\_button\_clicked",  
 triggered\_from:...

REQ-1772: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1773: export { CollectionInfoSidebar } from "./CollectionInfoSidebar";

REQ-1774: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1775: export { CollectionMenu } from "./CollectionMenu";

REQ-1776: export { default as CreateCollectionForm } from "./CreateCollectionForm";  
export type { CreateCollectionFormOwnProps } from "./CreateCollectionForm";

REQ-1777: export \* from "./MoveCollectionModal";

REQ-1778: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1779: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1780: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1781: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1782: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1783: export \* from "./TrashCollectionLanding";

REQ-1784: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1785: import type { CollectionItem } from "metabase-types/api";  
  
export const findLastEditedCollectionItem = (  
 collectionItems: CollectionItem[],  
) => {  
 return collectionItems.reduce((latest, item) =>...

REQ-1786: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1787: export \* from "./FormCollectionAndDashboardPicker";

REQ-1788: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1789: import getInitialCollectionId from "metabase/entities/collections/getInitialCollectionId";  
import { useSelector } from "metabase/lib/redux";  
import { PLUGIN\_COLLECTIONS } from...

REQ-1790: import type {  
 BookmarkId,  
 BookmarkType,  
 CardId,  
 Collection,  
 CollectionId,  
 CollectionItem,  
 Dashboard,  
 TableId,  
} from "metabase-types/api";  
import type { UploadMode } from...

REQ-1791: import { t } from "ttag";  
  
import { PLUGIN\_COLLECTIONS } from "metabase/plugins";  
import type {  
 Collection,  
 CollectionEssentials,  
 CollectionId,  
 CollectionItem,  
} from...

REQ-1792: import {  
 canonicalCollectionId,  
 getCollectionPathAsString,  
 isExamplesCollection,  
 isItemCollection,  
 isReadOnlyCollection,  
 isRootCollection,  
 isRootPersonalCollection,  
 ...

REQ-1793: export { AccordionList } from "./AccordionList";  
export type { Section, SearchProp, SearchProps } from "./types";

REQ-1794: import type { ReactNode } from "react";  
  
import type { IconName } from "metabase/ui";  
  
export type Item = object | string;  
  
export type Section<TItem extends Item = Item> = {  
 key?: string;  
 name?:...

REQ-1795: import { shallowEqual } from "@mantine/hooks";  
import { getIn } from "icepick";  
import { type ReactNode, isValidElement } from "react";  
import { isFragment } from "react-is";  
  
import type { Item,...

REQ-1796: import { getNextCursor, getPrevCursor } from "./utils";  
  
const canSelectSection = () => true;  
const cannotSelectSection = () => false;  
  
const sections = [  
 {  
 id: 1,  
 items: [  
 {  
 ...

REQ-1797: import { render } from "\_\_support\_\_/ui";  
import { delay } from "\_\_support\_\_/utils";  
import Question from "metabase-lib/v1/Question";  
import \* as ML\_Urls from "metabase-lib/v1/urls";  
  
import {...

REQ-1798: import type { ReactNode } from "react";  
  
export type AdminPaneProps = {  
 title?: React.ReactNode;  
 description?: string;  
 buttonText?: string;  
 buttonAction?: () => void;  
 buttonDisabled?:...

REQ-1799: export \* from "./AggregationPicker";

REQ-1800: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1801: export { AppBanner } from "./AppBanner";

REQ-1802: import dayjs from "dayjs";  
import utc from "dayjs/plugin/utc";  
  
dayjs.extend(utc);  
  
type Props = {  
 daysRemaining: number;  
 lastDismissed?: string | null;  
 tokenExpiryTimestamp: string;  
 ...

REQ-1803: import dayjs from "dayjs";  
  
import { getCurrentUTCTimestamp, shouldShowTrialBanner } from "./utils";  
  
describe("app banner utils", () => {  
 describe("shouldShowBanner", () => {  
 it("should return...

REQ-1804: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1805: export { Banner } from "./Banner";

REQ-1806: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1807: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1808: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1809: export const DEFAULT\_SIZE = 16;  
export const DEFAULT\_ICON\_PADDING = 4;  
export const DEFAULT\_CHECKED\_COLOR = "brand";  
export const DEFAULT\_UNCHECKED\_COLOR = "text-light";

REQ-1810: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1811: export interface CheckBoxInputProps {  
 size: number;  
}  
  
export interface CheckBoxContainerProps {  
 disabled: boolean | undefined;  
}  
  
export interface CheckBoxIconProps {  
 checked: boolean;  
 ...

REQ-1812: export function isEllipsisActive(node: HTMLElement): boolean {  
 return node.offsetWidth < node.scrollWidth;  
}

REQ-1813: export \* from "./CodeEditor";  
export type { CodeLanguage } from "./types";

REQ-1814: export type CodeLanguage =  
 | "clojure"  
 | "html"  
 | "json"  
 | "mustache"  
 | "pug"  
 | "python"  
 | "ruby"  
 | "typescript";

REQ-1815: import { html } from "@codemirror/lang-html";  
import { javascript } from "@codemirror/lang-javascript";  
import { json } from "@codemirror/lang-json";  
import { python } from...

REQ-1816: import { type Range, StateEffect, StateField } from "@codemirror/state";  
import { Decoration, EditorView } from "@codemirror/view";  
import { type RefObject, useEffect } from "react";  
  
import S from...

REQ-1817: export { CodeMirror, type CodeMirrorProps } from "./CodeMirror";  
export { type CodeMirrorRef } from "./types";

REQ-1818: export { type ReactCodeMirrorRef as CodeMirrorRef } from "@uiw/react-codemirror";

REQ-1819: import {  
 acceptCompletion,  
 moveCompletionSelection,  
 nextSnippetField,  
 prevSnippetField,  
} from "@codemirror/autocomplete";  
import { indentMore } from "@codemirror/commands";  
import {  
 ...

REQ-1820: eslint-disable-next-line import/no-default-export -- legacy usage

REQ-1821: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1822: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1823: export { ColorPill } from "./ColorPill";  
export type { PillSize } from "./types";

REQ-1824: export type PillSize = "small" | "medium";

REQ-1825: export { ColorRange } from "./ColorRange";

REQ-1826: export { ColorRangeSelector } from "./ColorRangeSelector";

REQ-1827: export { ColorSelector } from "./ColorSelector";

REQ-1828: export {  
 getLocalizationNoticeText,  
 CommunityLocalizationNotice,  
} from "./CommunityLocalizationNotice";

REQ-1829: export \* from "./CopyButton";

REQ-1830: export \* from "./CopyTextInput";

REQ-1831: export \* from "./DashboardSelector";

REQ-1832: export { DragDropContext } from "./DragDropContext";

REQ-1833: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1834: export { EditableDescription } from "./EditableDescription";

REQ-1835: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1836: export { Ellipsified } from "./Ellipsified";

REQ-1837: export { EllipsifiedPath } from "./EllipsifiedPath";

REQ-1838: export \* from "./EntityIdCard";

REQ-1839: import userEvent from "@testing-library/user-event";  
  
import { fireEvent, getIcon, render, screen } from "\_\_support\_\_/ui";  
import { delay } from "\_\_support\_\_/utils";  
import EntityMenuItem from...

REQ-1840: import { fireEvent, render, screen } from "\_\_support\_\_/ui";  
import EntityMenuTrigger from "metabase/common/components/EntityMenuTrigger";  
  
describe("EntityMenuTrigger", () => {  
 it("should render...

REQ-1841: export \* from "./AutoScrollBox";

REQ-1842: export \* from "./EntityPickerModal";

REQ-1843: export \* from "./ItemList";

REQ-1844: export \* from "./LoadingSpinner";

REQ-1845: export \* from "./NestedItemPicker";  
export \* from "./NestedItemPicker.styled";

REQ-1846: import type { PickerState } from "../../types";  
  
// reverse-traverse the statePath to find the last selected item  
export const findLastSelectedItem = <Item, Query>(  
 statePath: PickerState<Item,...

REQ-1847: export \* from "./RecentsTab";

REQ-1848: import dayjs from "dayjs";  
import relativeTime from "dayjs/plugin/relativeTime";  
import { t } from "ttag";  
dayjs.extend(relativeTime);  
  
import type { RecentItem } from "metabase-types/api";  
  
import...

REQ-1849: import dayjs from "dayjs";  
  
import { createMockRecentCollectionItem } from "metabase-types/api/mocks";  
  
import { getRecentGroups } from "./utils";  
  
const items = [  
 createMockRecentCollectionItem({  
...

REQ-1850: export \* from "./ResultItem";  
export \* from "./ResultItem.styled";

REQ-1851: export \* from "./SearchTab";

REQ-1852: export \* from "./AutoScrollBox";  
export \* from "./EntityPickerModal";  
export \* from "./ItemList";  
export \* from "./LoadingSpinner";  
export \* from "./NestedItemPicker";

REQ-1853: export const RECENTS\_TAB\_ID = "recents-tab";  
  
export const SEARCH\_TAB\_ID = "search-tab";

REQ-1854: export \* from "./use-log-recent-item";  
export \* from "./use-scoped-search-results";

REQ-1855: import { useMemo } from "react";  
  
import {  
 skipToken,  
 useGetDatabaseQuery,  
 useListCollectionItemsQuery,  
 useListDashboardItemsQuery,  
 useListDatabaseSchemaTablesQuery,  
} from...

REQ-1856: export \* from "./components";  
export \* from "./types";

REQ-1857: import type { ReactNode } from "react";  
  
import type { IconName } from "metabase/ui";  
import type { SearchResult, SearchResultId } from "metabase-types/api";  
  
import type { EntityPickerModalOptions }...

REQ-1858: import { c, msgid, t } from "ttag";  
  
import { color } from "metabase/lib/colors";  
import type { ObjectWithModel } from "metabase/lib/icon";  
import { getIcon } from "metabase/lib/icon";  
import {  
 ...

REQ-1859: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1860: export type ErrorDetails = string | Record<string, any>;  
  
export interface ErrorDetailsProps {  
 details?: ErrorDetails;  
 centered?: boolean;  
 className?: string;  
}

REQ-1861: import { trackSimpleEvent } from "metabase/lib/analytics";  
import type {  
 ErrorDiagnosticModalOpenedEvent,  
 ErrorDiagnosticModalSubmittedEvent,  
} from "metabase-types/analytics/event";  
  
export...

REQ-1862: export \* from "./ErrorPages";  
export \* from "./ErrorDiagnosticModal";

REQ-1863: import type {  
 Card,  
 Collection,  
 Dashboard,  
 DatasetData,  
 Log,  
 MetabaseInfo,  
} from "metabase-types/api";  
  
export type ReportableEntityName =  
 | "question"  
 | "model"  
 | "metric"  
 |...

REQ-1864: import { useAsync } from "react-use";  
import { t } from "ttag";  
  
import { getCurrentUser } from "metabase/admin/datamodel/selectors";  
import { useSelector } from "metabase/lib/redux";  
import {...

REQ-1865: import Bowser from "bowser";  
  
import { b64url\_to\_utf8 } from "metabase/lib/encoding";  
import { CardApi, CollectionsApi, DashboardApi } from "metabase/services";  
  
import type { ReportableEntityName }...

REQ-1866: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1867: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1868: export \* from "./ExportSettingsWidget";

REQ-1869: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1870: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1871: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1872: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1873: export type FieldAlignment = "start" | "end";  
  
export type FieldOrientation = "horizontal" | "vertical";

REQ-1874: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1875: export \* from "./FormFooter";

REQ-1876: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1877: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1878: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1879: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1880: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1881: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1882: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1883: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1884: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1885: export \* from "./InputWithSelectPrefix";

REQ-1886: export \* from "./BaseItemsTable";

REQ-1887: import type {  
 CollectionContentTableColumn,  
 CollectionContentTableColumnsMap,  
} from "metabase/collections/components/CollectionContent";  
import { type BreakpointName, breakpoints } from...

REQ-1888: import dayjs from "dayjs";  
import mockDate from "mockdate";  
  
import { renderWithProviders, screen } from "\_\_support\_\_/ui";  
import { createMockUser } from "metabase-types/api/mocks";  
  
import...

REQ-1889: export { LeaveRouteConfirmModal } from "./LeaveRouteConfirmModal";  
export { LeaveConfirmModal } from "./LeaveConfirmModal";

REQ-1890: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1891: import type { LinkProps as RouterLinkProps } from "react-router";  
  
import type { TooltipProps } from "metabase/ui";  
  
export interface LinkProps extends RouterLinkProps {  
 variant?: "default" |...

REQ-1892: import { Route } from "react-router";  
  
import { getIcon, renderWithProviders, screen } from "\_\_support\_\_/ui";  
  
import ListItem from "./ListItem";  
  
const ITEM\_NAME = "Table Foo";  
const...

REQ-1893: import { render, screen } from "\_\_support\_\_/ui";  
  
import ListSearchField from "./ListSearchField";  
  
describe("ListSearchField", () => {  
 it("should render", async () => {  
 render(<ListSearchField...

REQ-1894: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1895: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1896: export { MarkdownPreview } from "./MarkdownPreview";

REQ-1897: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1898: import { createMockEntitiesState } from "\_\_support\_\_/store";  
import { renderWithProviders, screen } from "\_\_support\_\_/ui";  
import { getMetadata } from "metabase/selectors/metadata";  
import {  
 ...

REQ-1899: import { setupFieldValuesEndpoint } from "\_\_support\_\_/server-mocks";  
import { createMockEntitiesState } from "\_\_support\_\_/store";  
import { renderWithProviders, screen } from "\_\_support\_\_/ui";  
import...

REQ-1900: export \* from "./ColumnFingerprintInfo";

REQ-1901: import { setupFieldsValuesEndpoints } from "\_\_support\_\_/server-mocks";  
import { createMockEntitiesState } from "\_\_support\_\_/store";  
import { renderWithProviders, screen } from...

REQ-1902: export { QueryColumnInfo, TableColumnInfo } from "./ColumnInfo";  
export type { QueryColumnInfoProps, TableColumnInfoProps } from "./ColumnInfo";

REQ-1903: export \* from "./ColumnInfoPopover";

REQ-1904: import { createMockEntitiesState } from "\_\_support\_\_/store";  
import { getIcon, renderWithProviders, screen } from "\_\_support\_\_/ui";  
import { getMetadata } from "metabase/selectors/metadata";  
import...

REQ-1905: export { SemanticTypeLabel } from "./SemanticTypeLabel";

REQ-1906: export type { TableInfoProps } from "./TableInfo";  
// eslint-disable-next-line import/no-default-export -- deprecated usage  
export { default } from "./TableInfo";

REQ-1907: export \* from "./TableInfoPopover";

REQ-1908: eslint-disable-next-line no-restricted-imports

REQ-1909: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1910: import type { ReactNode } from "react";  
  
export interface CommonModalProps {  
 // takes over the entire screen  
 fullPageModal?: boolean;  
 // standard modal  
 formModal?: boolean;  
 centeredTitle?:...

REQ-1911: import {  
 skipToken,  
 useListCollectionDashboardQuestionCandidatesQuery,  
} from "metabase/api";  
import { useSelector } from "metabase/lib/redux";  
import { getUserIsAdmin } from...

REQ-1912: export \* from "./MultiContainerDraggableContext";

REQ-1913: import { trackSimpleEvent } from "metabase/lib/analytics";  
  
export const trackNewMenuItemClicked = (  
 item: "question" | "native-query" | "dashboard",  
) =>  
 trackSimpleEvent({  
 event:...

REQ-1914: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1915: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1916: import { fireEvent, render, screen } from "\_\_support\_\_/ui";  
import PasswordReveal from "metabase/common/components/PasswordReveal";  
  
describe("password reveal", () => {  
 it("should toggle the...

REQ-1917: export \* from "./CollectionPicker";  
export \* from "./CollectionPickerModal";

REQ-1918: import { useEffect, useMemo, useState } from "react";  
import { t } from "ttag";  
  
import {  
 skipToken,  
 useGetCollectionQuery,  
 useListCollectionItemsQuery,  
} from "metabase/api";  
import {...

REQ-1919: export \* from "./components";  
export \* from "./hooks";  
export \* from "./types";

REQ-1920: import type {  
 CardId,  
 CollectionId,  
 CollectionItemModel,  
 DashboardId,  
 ListCollectionItemsRequest,  
 SearchResult,  
} from "metabase-types/api";  
  
import type {  
 EntityPickerModalOptions,  
 ...

REQ-1921: import type {  
 CollectionId,  
 CollectionItemModel,  
 ListCollectionItemsRequest,  
} from "metabase-types/api";  
  
import type { PickerState } from "../../EntityPicker";  
import type {...

REQ-1922: import { getParentCollectionId } from "./utils";  
  
describe("CollectionPicker > utils", () => {  
 describe("getParentCollectionId", () => {  
 it("should get the root collection for null values", ()...

REQ-1923: export \* from "./DashboardPicker";  
export \* from "./DashboardPickerModal";

REQ-1924: export \* from "./components";  
export \* from "./types";

REQ-1925: import type {  
 CollectionId,  
 DashboardId,  
 ListCollectionItemsRequest,  
 SearchModel,  
} from "metabase-types/api";  
  
import type {  
 EntityPickerModalOptions,  
 ListProps,  
 PickerState,  
} from...

REQ-1926: import \_ from "underscore";  
  
import type { CollectionId, Dashboard } from "metabase-types/api";  
  
import {  
 getParentCollectionId,  
 getPathLevelForItem,  
} from "../CollectionPicker/utils";  
  
import...

REQ-1927: export \* from "./DataPickerModal";

REQ-1928: export \* from "./useAvailableData";

REQ-1929: import { useSearchQuery } from "metabase/api";  
import type { DatabaseId } from "metabase-types/api";  
  
interface Props {  
 databaseId?: DatabaseId;  
}  
  
export const useAvailableData = ({ databaseId }:...

REQ-1930: export \* from "./components";  
export \* from "./types";  
export \* from "./utils";

REQ-1931: import type {  
 CardId,  
 Collection,  
 DashboardId,  
 DatabaseId,  
 SchemaName,  
 TableId,  
} from "metabase-types/api";  
  
import type { EntityPickerModalOptions } from "../../EntityPicker";  
import...

REQ-1932: import { humanize, titleize } from "metabase/lib/formatting";  
import { isNullOrUndefined } from "metabase/lib/types";  
import \* as Lib from "metabase-lib";  
import { getSchemaName } from...

REQ-1933: export \* from "./QuestionPicker";  
export \* from "./QuestionPickerModal";

REQ-1934: export \* from "./components";  
export \* from "./types";  
export { getQuestionPickerValue } from "./utils";

REQ-1935: import type {  
 CardId,  
 ListCollectionItemsRequest,  
 SearchModel,  
} from "metabase-types/api";  
  
import type {  
 EntityPickerModalOptions,  
 ListProps,  
 PickerState,  
} from...

REQ-1936: import \_ from "underscore";  
  
import type { Card, CardType, CollectionItemModel } from "metabase-types/api";  
  
import type {  
 QuestionPickerItem,  
 QuestionPickerValue,  
 QuestionPickerValueModel,  
}...

REQ-1937: import {  
 skipToken,  
 useGetCardQuery,  
 useGetCollectionQuery,  
 useGetDashboardQuery,  
} from "metabase/api";  
import { isValidCollectionId } from "metabase/collections/utils";  
  
import type {...

REQ-1938: import { PERSONAL\_COLLECTIONS } from "metabase/entities/collections/constants";  
import { isNullOrUndefined } from "metabase/lib/types";  
import type {  
 CollectionId,  
 CollectionItemModel,  
 ...

REQ-1939: import { getCollectionIdPath } from "./utils";  
  
describe("getCollectionIdPath", () => {  
 it("should handle the current user's personal collection", () => {  
 const path = getCollectionIdPath(  
 ...

REQ-1940: import \* as popper from "@popperjs/core";  
  
const PAGE\_PADDING = 10;  
const SIZE\_TO\_FIT\_MIN\_HEIGHT = 200;  
  
export type SizeToFitOptions = {  
 minHeight: number;  
};  
  
export function...

REQ-1941: eslint-disable react/prop-types

REQ-1942: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1943: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1944: export \* from "./BucketPickerPopover";

REQ-1945: import type \* as Lib from "metabase-lib";  
  
import type { BaseBucketPickerPopoverProps } from "./BaseBucketPickerPopover";  
  
type CommonProps = Pick<  
 BaseBucketPickerPopoverProps,  
 | "query"  
 |...

REQ-1946: export \* from "./QueryColumnPicker";

REQ-1947: import { setupCardQueryEndpoints } from "\_\_support\_\_/server-mocks";  
import { render } from "\_\_support\_\_/ui";  
import Question from "metabase-lib/v1/Question";  
import { createMockDataset } from...

REQ-1948: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1949: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1950: export type RadioVariant = "normal" | "underlined" | "bubble";  
export type RadioColorScheme = "default" | "accent7";

REQ-1951: export \* from "./SaveQuestionTitle";  
export \* from "./SaveQuestionForm";

REQ-1952: import \* as Yup from "yup";  
  
import \* as Errors from "metabase/lib/errors";  
import { QUESTION\_NAME\_MAX\_LENGTH } from "metabase/questions/constants";  
  
export const SAVE\_QUESTION\_SCHEMA = Yup.object({  
...

REQ-1953: import type Question from "metabase-lib/v1/Question";  
import type {  
 CollectionId,  
 DashboardId,  
 DashboardTabId,  
} from "metabase-types/api";  
  
export type SaveQuestionProps<C = CollectionId> = {  
...

REQ-1954: import { P, match } from "ts-pattern";  
import { t } from "ttag";  
  
import { canonicalCollectionId } from "metabase/collections/utils";  
import { isNullOrUndefined } from "metabase/lib/types";  
import...

REQ-1955: import Question from "metabase-lib/v1/Question";  
import { createMockCard } from "metabase-types/api/mocks";  
  
import type { CreateQuestionOptions } from "./types";  
import {  
 createQuestion,  
 ...

REQ-1956: export \* from "./SaveQuestionModal";

REQ-1957: import {  
 setupCardEndpoints,  
 setupCardQueryMetadataEndpoint,  
 setupDatabaseEndpoints,  
 setupUnauthorizedCardEndpoints,  
 setupUnauthorizedSchemaEndpoints,  
} from...

REQ-1958: export const defaultDay = "mon";  
export const defaultHour = 8;

REQ-1959: import { c, msgid, ngettext, t } from "ttag";  
import \_ from "underscore";  
  
import { has24HourModeSetting } from "metabase/lib/time";  
import type { ScheduleDayType, ScheduleFrameType } from...

REQ-1960: import { has24HourModeSetting } from "metabase/lib/time";  
  
import { getHours } from "./strings";  
  
jest.mock("metabase/lib/time", () => ({  
 has24HourModeSetting: jest.fn(),  
}));  
  
describe("getHours",...

REQ-1961: import type { ScheduleSettings, ScheduleType } from "metabase-types/api";  
  
type ScheduleProperty = keyof ScheduleSettings;  
export type ScheduleChangeProp = { name: ScheduleProperty; value: unknown...

REQ-1962: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1963: export \* from "./SegmentedControl";

REQ-1964: import { render, screen } from "\_\_support\_\_/ui";  
import Select, { Option } from "metabase/common/components/Select";  
  
describe("Select", () => {  
 it("should render selected option", () => {  
 ...

REQ-1965: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1966: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1967: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1968: export { InsightsTab } from "./InsightsTab";

REQ-1969: export { InsightsTabOrLink } from "./InsightsTabOrLink";

REQ-1970: export \* from "./Sidesheet";  
export \* from "./SidesheetButton";  
export \* from "./SidesheetCard";  
export \* from "./SidesheetCardSection";  
export \* from "./SidesheetSubPage";  
export \* from...

REQ-1971: export \* from "./Sortable";  
export \* from "./SortableList";

REQ-1972: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1973: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1974: export function getTabId<T>(idPrefix: string, value: T): string {  
 return `${idPrefix}-T-${value}`;  
}  
  
export function getTabPanelId<T>(idPrefix: string, value: T): string {  
 return...

REQ-1975: export \* from "./TabButton";

REQ-1976: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1977: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1978: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1979: import type { CollisionDetection } from "@dnd-kit/core";  
  
export const tabsCollisionDetection: CollisionDetection = ({  
 active,  
 collisionRect,  
 droppableRects,  
 droppableContainers,  
}) => {  
 //...

REQ-1980: export \* from "./TabRow";

REQ-1981: export \* from "./ClientSortableTable";  
export \* from "./Table";

REQ-1982: export type BaseRow = Record<string, any> & { id: number | string };  
  
export type ColumnItem = {  
 name: string;  
 key: string;  
 sortable?: boolean;  
};

REQ-1983: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1984: export \* from "./Timeline";

REQ-1985: import { t } from "ttag";  
  
import type { Revision, User } from "metabase-types/api";  
  
export function getTimelineEvents({  
 revisions = [],  
 currentUser,  
}: {  
 revisions: Revision[] | undefined;  
 ...

REQ-1986: export { TitleAndDescription } from "./TitleAndDescription";

REQ-1987: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1988: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1989: eslint-disable react/prop-types

REQ-1990: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1991: export function parseStringValue(value: any): string | null {  
 const trimmedValue = trim(value);  
 if (trimmedValue === "") {  
 return null;  
 }  
  
 return trimmedValue;  
}  
  
function trim(value:...

REQ-1992: import { parseStringValue } from "./utils";  
  
describe("metabase/common/components/TokenField/utils", () => {  
 describe("parseStringValue", () => {  
 it("should return null for falsy and whitespace...

REQ-1993: eslint-disable-next-line no-restricted-imports

REQ-1994: export \* from "./TokenFieldItem.styled";

REQ-1995: export \* from "./ToolbarButton";

REQ-1996: import userEvent from "@testing-library/user-event";  
import { forwardRef, useState } from "react";  
  
import { render, screen } from "\_\_support\_\_/ui";  
  
import Tooltip from "./Tooltip";  
  
const...

REQ-1997: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1998: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-1999: export \* from "./ViewFooterButton";

REQ-2000: export \* from "./VariableHeightVirtualizedList";

REQ-2001: NOTE: we currently use object's `model` property for the drag type

REQ-2002: export \* from "./Tree";

REQ-2003: import type \* as React from "react";  
  
import type { IconName, IconProps } from "metabase/ui";  
  
export interface ITreeNodeItem {  
 id: string | number;  
 name: string;  
 icon: IconName | IconProps;  
 ...

REQ-2004: export const DEFAULT\_UPLOAD\_INPUT\_ID = "upload-input";

REQ-2005: export \* from "./UploadTooltip";  
export \* from "./UploadInput";  
export \* from "./UploadLabel";

REQ-2006: export const LOAD\_COMPLETE\_FAVICON = "app/assets/img/blue\_check.png";

REQ-2007: /\*  
 \* This directory is deprecated.  
 \* Use "metabase/api" instead.  
 \*/  
  
export \* from "./use-bookmark-list-query";  
export \* from "./use-collection-query";  
export \* from...

REQ-2008: export \* from "./use-bookmark-list-query";

REQ-2009: import Bookmarks from "metabase/entities/bookmarks";  
import type { Bookmark } from "metabase-types/api";  
  
import type {  
 UseEntityListQueryProps,  
 UseEntityListQueryResult,  
} from...

REQ-2010: export \* from "./use-collection-query";

REQ-2011: import Collections from "metabase/entities/collections";  
import type { Collection, CollectionId } from "metabase-types/api";  
  
import type {  
 UseEntityQueryProps,  
 UseEntityQueryResult,  
} from...

REQ-2012: export \* from "./use-dashboard-query";

REQ-2013: import type {  
 UseEntityQueryProps,  
 UseEntityQueryResult,  
} from "metabase/common/hooks/entity-framework/use-entity-query";  
import { useEntityQuery } from...

REQ-2014: export \* from "./use-database-list-query";

REQ-2015: import type {  
 UseEntityListQueryProps,  
 UseEntityListQueryResult,  
} from "metabase/common/hooks/entity-framework/use-entity-list-query";  
import { useEntityListQuery } from...

REQ-2016: export \* from "./use-database-query";

REQ-2017: import type {  
 UseEntityQueryProps,  
 UseEntityQueryResult,  
} from "metabase/common/hooks/entity-framework/use-entity-query";  
import { useEntityQuery } from...

REQ-2018: export \* from "./use-entity-list-query";

REQ-2019: import type { Action } from "@reduxjs/toolkit";  
import { useDeepCompareEffect, usePrevious } from "react-use";  
  
import { useDispatch, useSelector } from "metabase/lib/redux";  
import type { State }...

REQ-2020: export \* from "./use-entity-query";

REQ-2021: import type { Action } from "@reduxjs/toolkit";  
import { useDeepCompareEffect } from "react-use";  
  
import { useDispatch, useSelector } from "metabase/lib/redux";  
import type { State } from...

REQ-2022: export \* from "./use-group-list-query";

REQ-2023: import type {  
 UseEntityListQueryProps,  
 UseEntityListQueryResult,  
} from "metabase/common/hooks/entity-framework/use-entity-list-query";  
import { useEntityListQuery } from...

REQ-2024: export \* from "./use-question-list-query";

REQ-2025: import type {  
 UseEntityListQueryProps,  
 UseEntityListQueryResult,  
} from "metabase/common/hooks/entity-framework/use-entity-list-query";  
import { useEntityListQuery } from...

REQ-2026: export \* from "./use-question-query";

REQ-2027: import type {  
 UseEntityQueryProps,  
 UseEntityQueryResult,  
} from "metabase/common/hooks/entity-framework/use-entity-query";  
import { useEntityQuery } from...

REQ-2028: export \* from "./use-revision-list-query";

REQ-2029: import type {  
 UseEntityListQueryProps,  
 UseEntityListQueryResult,  
} from "metabase/common/hooks/entity-framework/use-entity-list-query";  
import { useEntityListQuery } from...

REQ-2030: export \* from "./use-search-list-query";

REQ-2031: import Search from "metabase/entities/search";  
import type {  
 CollectionItem,  
 SearchRequest,  
 SearchResponse,  
} from "metabase-types/api";  
  
import type {  
 UseEntityListQueryProps,  
 ...

REQ-2032: export \* from "./use-table-list-query";

REQ-2033: import type {  
 UseEntityListQueryProps,  
 UseEntityListQueryResult,  
} from "metabase/common/hooks/entity-framework/use-entity-list-query";  
import { useEntityListQuery } from...

REQ-2034: export \* from "./use-table-query";

REQ-2035: import type {  
 UseEntityQueryProps,  
 UseEntityQueryResult,  
} from "metabase/common/hooks/entity-framework/use-entity-query";  
import { useEntityQuery } from...

REQ-2036: export \* from "./entity-framework";  
export \* from "./use-confirmation";  
export \* from "./use-docs-url";  
export \* from "./use-has-token-feature";  
export \* from "./use-locale";  
export \* from...

REQ-2037: export \* from "./use-action-button-label";

REQ-2038: import { type ReactNode, useRef, useState } from "react";  
  
interface UseActionButtonLabelProps {  
 defaultLabel: string | ReactNode;  
 timeout?: number;  
}  
  
/\*\*  
 \* Small hook to temporarly update a...

REQ-2039: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2040: import { useBeforeUnload as useBeforeUnloadHook } from "react-use";  
import { t } from "ttag";  
  
// most browsers don't use a custom message with beforeunload anymore, just putting here to retain...

REQ-2041: export {  
 useCallbackEffect,  
 type ScheduleCallback,  
} from "./use-callback-effect";

REQ-2042: import { useCallback, useEffect, useState } from "react";  
  
type Callback = () => void | Promise<void>;  
  
type IsScheduled = boolean;  
  
export type ScheduleCallback = (callback: Callback) => void;  
  
/\*\*  
...

REQ-2043: import type { Location } from "history";  
import { useCallback, useEffect, useState } from "react";  
import type { InjectedRouter, Route } from "react-router";  
import { goBack, push, replace } from...

REQ-2044: import { useRef } from "react";  
  
export const useCurrentRef = function <T>(value: T) {  
 const ref = useRef(value);  
 ref.current = value;  
  
 return ref;  
};

REQ-2045: import { useEffect, useState } from "react";  
  
export function useDebouncedValue<TVALUE>(  
 value: TVALUE,  
 delay: number,  
 onlyFor: (lastValue: TVALUE, newValue: TVALUE) => boolean = () => true,  
):...

REQ-2046: export \* from "./use-docs-url";

REQ-2047: import { useSelector } from "metabase/lib/redux";  
import {  
 type UtmProps,  
 getDocsUrl,  
 getLearnUrl,  
} from "metabase/selectors/settings";  
import { getShowMetabaseLinks } from...

REQ-2048: export \* from "./use-escape-to-close-modal";

REQ-2049: import { useEffect } from "react";  
  
import { useSetting } from "metabase/common/hooks";  
  
export const useFavicon = ({ favicon }: { favicon: string | null }) => {  
 const defaultFavicon =...

REQ-2050: import { useReducer } from "react";  
  
// https://reactjs.org/docs/hooks-faq.html#is-there-something-like-forceupdate  
export function useForceUpdate() {  
 const [, forceUpdate] = useReducer((x) => x +...

REQ-2051: export \* from "./use-has-token-feature";

REQ-2052: import { useSelector } from "metabase/lib/redux";  
import { getTokenFeature } from "metabase/setup/selectors";  
import type { TokenFeature } from "metabase-types/api";  
  
export const useHasTokenFeature...

REQ-2053: export \* from "./use-instance-locale";

REQ-2054: import { useSelector } from "metabase/lib/redux";  
import { getSetting } from "metabase/selectors/settings";  
  
export const useInstanceLocale = () => {  
 return useSelector((state) => getSetting(state,...

REQ-2055: import { useLocation } from "react-use";  
  
import { useSelector } from "metabase/lib/redux";  
import { getCustomHomePageDashboardId } from "metabase/selectors/app";  
  
import { useSetting } from...

REQ-2056: import { useMedia } from "react-use";  
  
const useIsSmallScreen = () => {  
 return useMedia("(max-width: 40em)");  
};  
  
// eslint-disable-next-line import/no-default-export -- deprecated usage  
export...

REQ-2057: import { useLayoutEffect, useRef, useState } from "react";  
import \_ from "underscore";  
  
import resizeObserver from "metabase/lib/resize-observer";  
  
type UseIsTruncatedProps = {  
 disabled?:...

REQ-2058: import { useEffect } from "react";  
  
export function useKeyboardShortcut(  
 key: string,  
 callback: (e: KeyboardEvent) => void,  
) {  
 useEffect(() => {  
 function keyboardListener(e: KeyboardEvent)...

REQ-2059: export \* from "./use-list-keyboard-navigation";

REQ-2060: import type { MutableRefObject } from "react";  
import { useCallback, useEffect, useRef, useState } from "react";  
  
interface ListKeyboardNavigationInput<T, R> {  
 ref?: MutableRefObject<R | null>;  
 ...

REQ-2061: import { renderHook } from "@testing-library/react";  
  
import { fireEvent } from "\_\_support\_\_/ui";  
  
import { useListKeyboardNavigation } from "./use-list-keyboard-navigation";  
  
const list = [  
 { id:...

REQ-2062: import { useCallback, useState } from "react";  
  
export type UseListSelectReturnValue<T> = {  
 clear: () => void;  
 getIsSelected: (item: T) => boolean;  
 selected: T[];  
 selectOnlyTheseItems:...

REQ-2063: import { act, renderHook } from "@testing-library/react";  
  
import { useListSelect } from "./use-list-select";  
  
interface objectType {  
 id: number;  
 name: string;  
}  
  
const OBJECT\_LIST = [  
 { id: 1,...

REQ-2064: import { useEffect } from "react";  
  
interface LoadingTimerProps {  
 timer: number;  
 onTimeout: () => void;  
}  
  
export function useLoadingTimer(isLoading: boolean, props: LoadingTimerProps) {  
 const...

REQ-2065: export \* from "./use-locale";

REQ-2066: import { useContext } from "react";  
  
import { getCurrentUser } from "metabase/admin/datamodel/selectors";  
import { useInstanceLocale } from "metabase/common/hooks/use-instance-locale";  
import {...

REQ-2067: import { useMemo } from "react";  
  
type MapLike<K, V> = Map<K, V> | WeakMap<object & K, V>;  
  
function getWithFallback<K, V>(  
 map: MapLike<K, V>,  
 key: K,  
 fallback: () => V,  
): V {  
 if ("has" in...

REQ-2068: import { memoize } from "./use-memoized-callback";  
  
describe("memoize", () => {  
 it("should return same result for same arguments", () => {  
 const add = jest.fn((a: number, b: number) => a + b);  
...

REQ-2069: import { useEffect, useState } from "react";  
  
// this is a custom hook that is used to open a modal after a delay  
// so that the modal can be animated in  
export function useModalOpen() {  
 const...

REQ-2070: import { useCallback, useRef } from "react";  
  
type AsyncFn = (...args: any[]) => Promise<any>;  
  
type AsyncReturnType<T extends (...args: any) => Promise<any>> = T extends (  
 ...args: any  
) =>...

REQ-2071: export \* from "./use-notification-channels";

REQ-2072: import { useGetChannelInfoQuery } from "metabase/api";  
  
export const useHasAnyNotificationChannel = (): boolean => {  
 const { data: channelInfo } = useGetChannelInfoQuery();  
  
 return...

REQ-2073: import { useCallback } from "react";  
  
import { useSetting } from "metabase/common/hooks";  
import {  
 type FormatNumberOptions,  
 formatNumber,  
} from "metabase/lib/formatting/numbers";  
  
export type...

REQ-2074: import { mockSettings } from "\_\_support\_\_/settings";  
import { renderHookWithProviders } from "\_\_support\_\_/ui";  
import type { NumberFormattingSettings } from "metabase-types/api";  
  
import {  
 type...

REQ-2075: import type { RefObject } from "react";  
import { useEffect } from "react";  
  
interface ValidRefTarget {  
 contains(target: EventTarget | null): boolean;  
}  
  
export function useOnClickOutside<T extends...

REQ-2076: import { useCallback, useState } from "react";  
  
export const usePagination = (initialPage = 0) => {  
 const [page, setPage] = useState(initialPage);  
  
 const handleNextPage = useCallback(  
 () =>...

REQ-2077: import { act, renderHook } from "@testing-library/react";  
  
import { usePagination } from "./use-pagination";  
  
describe("usePagination", () => {  
 it("should set 'page' to 'initialPage' upon calling...

REQ-2078: import { useMemo } from "react";  
  
import type { ColorPalette } from "metabase/lib/colors/types";  
import { useMantineTheme } from "metabase/ui";  
  
/\*\*  
 \* Extracts a color palette from a subset of...

REQ-2079: import { useCallback } from "react";  
import { useMountedState } from "react-use";  
  
type AsyncFn = (...args: any[]) => Promise<any>;  
  
/\*\*  
 \* wraps the given async function in a promise that does not...

REQ-2080: import { useEffect, useRef } from "react";  
  
export const useScrollOnMount = () => {  
 const ref = useRef(null);  
  
 useEffect(() => {  
 if (ref.current) {  
 ref.current.scrollIntoView?.({ block:...

REQ-2081: import { useCallback, useRef } from "react";  
import \_ from "underscore";  
  
import { isElement } from "metabase-types/guards";  
  
type PopoverData = {  
 contentEl: Element;  
 backdropEl?: Element;  
 ...

REQ-2082: export \* from "./use-setting";

REQ-2083: import { useCallback, useMemo } from "react";  
import \_ from "underscore";  
  
import { useDispatch, useSelector } from "metabase/lib/redux";  
import { updateUserSetting } from...

REQ-2084: import { useSelector } from "metabase/lib/redux";  
import { type StorePaths, getStoreUrl } from "metabase/selectors/settings";  
  
export function useStoreUrl(storePath: StorePaths = "") {  
 return...

REQ-2085: export \* from "./use-temp-storage";

REQ-2086: import { useCallback } from "react";  
  
import { useDispatch, useSelector } from "metabase/lib/redux";  
import { setTempSetting } from "metabase/redux/app";  
import type {  
 State,  
 TempStorageKey,  
 ...

REQ-2087: import { useCallback, useRef, useState } from "react";  
import { useUnmount } from "react-use";  
  
/\*\*  
 \* A hook that temporarily changes the value of a stateful value  
 \* and automatically resets it to...

REQ-2088: export \* from "./use-toast";

REQ-2089: import { useCallback } from "react";  
  
import { useDispatch } from "metabase/lib/redux";  
import { addUndo, dismissUndo } from "metabase/redux/undo";  
import type { Undo } from...

REQ-2090: import { useCallback, useState } from "react";  
  
type ToggleHookResult = [  
 boolean,  
 {  
 turnOn: () => void;  
 turnOff: () => void;  
 toggle: () => void;  
 },  
];  
  
/\*\*  
 \* @deprecated use...

REQ-2091: import { useRef } from "react";  
import \_ from "underscore";  
  
export const useUniqueId = (prefix?: string): string => {  
 const idRef = useRef("");  
 if (!idRef.current) {  
 idRef.current =...

REQ-2092: import { useLayoutEffect, useRef } from "react";  
  
// identical to useUnmount from react-use but leverages useLayoutEffect  
// instead of useEffect in the case you need access to the DOM elements  
//...

REQ-2093: export \* from "./types";  
export \* from "./use-url-state";  
export \* from "./utils";

REQ-2094: import type { Query } from "history";  
  
export type QueryParam = Query[keyof Query];

REQ-2095: import type { Location, Query } from "history";  
import { useCallback, useEffect, useState } from "react";  
import { push, replace } from "react-router-redux";  
import { useEffectOnce, useLatest } from...

REQ-2096: import type { Location } from "history";  
  
import { act, renderHookWithProviders, waitFor } from "\_\_support\_\_/ui";  
import { createMockLocation } from "metabase-types/store/mocks";  
  
import type {...

REQ-2097: import type { QueryParam } from "./types";  
  
export function getFirstParamValue(param: QueryParam) {  
 return Array.isArray(param) ? param[0] : param;  
}

REQ-2098: export { useUrlWithUtm } from "./use-url-with-utm";

REQ-2099: import { useSelector } from "metabase/lib/redux";  
import { type UtmProps, getUrlWithUtm } from "metabase/selectors/settings";  
  
export function useUrlWithUtm(url: string, utm: UtmProps):...

REQ-2100: import { useCallback } from "react";  
  
import { useUserKeyValue } from "./use-user-key-value";  
  
type UseUserAcknowledgementResult = [  
 acknowledged: boolean,  
 { ack: () => void; unack: () => void;...

REQ-2101: import { useCallback } from "react";  
  
import {  
 skipToken,  
 useDeleteUserKeyValueMutation,  
 useGetUserKeyValueQuery,  
 useUpdateKeyValueMutation,  
} from "metabase/api";  
import { useSelector } from...

REQ-2102: import { useCallback } from "react";  
  
const hasNotificationAPI = "Notification" in window;  
  
export function useWebNotification() {  
 const requestPermission = useCallback(async () => {  
 if...

REQ-2103: eslint-disable-next-line no-restricted-imports

REQ-2104: export type InputSize = "small" | "medium" | "large";

REQ-2105: export type TableExportFormat = "csv" | "xlsx" | "json";  
export type ExportFormat = TableExportFormat | "png";

REQ-2106: export type PartialBy<T, K extends keyof T> = Omit<T, K> & Partial<Pick<T, K>>;

REQ-2107: import type { IconName } from "metabase/ui";  
import type { ColumnGroupDisplayInfo } from "metabase-lib";  
  
export function getColumnGroupIcon(  
 groupInfo: ColumnGroupDisplayInfo,  
): IconName {  
 if...

REQ-2108: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import { checkNotNull } from "metabase/lib/types";  
import \* as Lib from "metabase-lib";  
import { SAMPLE\_METADATA } from...

REQ-2109: import type { IconName } from "metabase/ui";  
import \* as Lib from "metabase-lib";  
  
export function getColumnIcon(  
 column: Lib.ColumnMetadata | Lib.ColumnTypeInfo,  
): IconName {  
 if...

REQ-2110: import \* as Lib from "metabase-lib";  
import { columnFinder, createQuery } from "metabase-lib/test-helpers";  
  
import { getColumnIcon } from "./columns";  
  
describe("common/utils/columns", () => {  
 ...

REQ-2111: export const doNotForwardProps = (...propNamesToBlock: string[]) => ({  
 shouldForwardProp: (propName: string) => !propNamesToBlock.includes(propName),  
});

REQ-2112: \* Returns true if e.key is the given key and no modifier keys (ctrl, meta, alt, shift) were pressed

REQ-2113: import { t } from "ttag";  
  
const TRANSLATED\_NAME\_BY\_MODEL\_TYPE: Record<string, string> = {  
 get action() {  
 return t`Action`;  
 },  
 get card() {  
 return t`Question`;  
 },  
 get collection()...

REQ-2114: import { getTranslatedEntityName } from "./model-names";  
  
describe("common/utils/model-names", () => {  
 it("returns null if the model doesn't exist", () => {  
 ...

REQ-2115: import type { TokenFeatures } from "metabase-types/api";  
import { tokenFeatures } from "metabase-types/api";  
  
export type Plan =  
 | "oss"  
 | "starter"  
 | "starter-with-dwh"  
 | "pro-cloud"  
 |...

REQ-2116: import { getPlan } from "metabase/common/utils/plan";  
import { createMockTokenFeatures } from "metabase-types/api/mocks";  
  
describe("common/utils/columns", () => {  
 describe("getPlan", () => {  
 ...

REQ-2117: eslint-disable-next-line no-restricted-imports

REQ-2118: eslint-disable-next-line no-restricted-imports

REQ-2119: export const OVERLAY\_Z\_INDEX = 200;  
  
/\*\* This constant is used to work around a bug in Mantine: when certain Mantine  
 \* overlays appear above a Mantine Modal, only the Modal's portal has  
 \*...

REQ-2120: import {  
 getActionErrorMessage,  
 getActionExecutionMessage,  
} from "metabase/actions/utils";  
import { SIDEBAR\_NAME } from "metabase/dashboard/constants";  
import { addUndo } from...

REQ-2121: import { DashboardApi } from "metabase/services";  
import { createMockDashboard } from "metabase-types/api/mocks";  
import {  
 createMockDashboardState,  
 createMockRoutingState,  
 createMockState,  
}...

REQ-2122: import {  
 closeAutoWireParameterToast,  
 showAddedCardAutoWireParametersToast,  
 showAutoWireParametersToast,  
} from "metabase/dashboard/actions/auto-wire-parameters/toasts";  
import {  
 ...

REQ-2123: export const AUTO\_WIRE\_TOAST\_TIMEOUT = 12000;  
export const AUTO\_WIRE\_UNDO\_TOAST\_TIMEOUT = 8000;

REQ-2124: import { t } from "ttag";  
import \_ from "underscore";  
  
import type { SetMultipleDashCardAttributesOpts } from "metabase/dashboard/actions";  
import {  
 setDashCardAttributes,  
 ...

REQ-2125: import \_ from "underscore";  
  
import { isActionDashCard } from "metabase/actions/utils";  
import { getExistingDashCards } from "metabase/dashboard/actions/utils";  
import {  
 ...

REQ-2126: import { createAction } from "@reduxjs/toolkit";  
import { t } from "ttag";  
import \_ from "underscore";  
  
import Questions from "metabase/entities/questions";  
import {  
 DEFAULT\_CARD\_SIZE,  
 ...

REQ-2127: import { t } from "ttag";  
  
import { createCard } from "metabase/lib/card";  
  
import { trackCardCreated } from "../analytics";  
  
import { addDashCardToDashboard } from "./cards-typed";  
  
export const...

REQ-2128: import type { Store } from "@reduxjs/toolkit";  
import \_ from "underscore";  
  
import { getStore } from "\_\_support\_\_/entities-store";  
import {  
 setupCardQueryEndpoints,  
 ...

REQ-2129: import { createAction } from "@reduxjs/toolkit";  
import { push } from "react-router-redux";  
  
import { getLocation } from "metabase/selectors/routing";  
import type {  
 DashCardId,  
 ...

REQ-2130: import { createAction } from "@reduxjs/toolkit";  
import { getIn } from "icepick";  
import { denormalize, normalize, schema } from "normalizr";  
import { match } from "ts-pattern";  
import { t } from...

REQ-2131: import { getStore } from "\_\_support\_\_/entities-store";  
import {  
 setupDashboardQueryMetadataEndpoint,  
 setupDashboardsEndpoints,  
 setupDatabaseEndpoints,  
} from "\_\_support\_\_/server-mocks";  
import...

REQ-2132: import \_ from "underscore";  
  
import { getCardAfterVisualizationClick } from "metabase/visualizations/lib/utils";  
import \* as Lib from "metabase-lib";  
import Question from...

REQ-2133: import {  
 createMockCard,  
 createMockDashboard,  
 createMockDashboardCard,  
 createMockParameter,  
} from "metabase-types/api/mocks";  
  
import { getParametersMappedToCard } from...

REQ-2134: export \* from "./cards-typed";  
export \* from "./cards";  
export \* from "./core";  
export \* from "./data-fetching";  
export \* from "./navigation";  
export \* from "./parameters";  
export \* from...

REQ-2135: import { createThunkAction } from "metabase/lib/redux";  
import \* as Urls from "metabase/lib/urls";  
import { openUrl } from "metabase/redux/app";  
import { getMetadata } from...

REQ-2136: import { getStore } from "\_\_support\_\_/entities-store";  
import { getParameters } from "metabase/dashboard/selectors";  
import { mainReducers } from "metabase/reducers-main";  
import {  
 createMockCard,  
...

REQ-2137: import {  
 fetchDashboard,  
 fetchDashboardCardData,  
} from "metabase/dashboard/actions";  
import Revisions from "metabase/entities/revisions";  
import { createThunkAction } from...

REQ-2138: import { assocIn, dissocIn, getIn } from "icepick";  
import \_ from "underscore";  
  
import {  
 fetchDashboard,  
 fetchDashboardCardData,  
} from "metabase/dashboard/actions";  
import Dashboards from...

REQ-2139: import { SIDEBAR\_NAME } from "metabase/dashboard/constants";  
import type { Dispatch } from "metabase-types/store";  
  
import { closeSidebar, setSidebar } from "./ui";  
  
export const setSharing =...

REQ-2140: import { arrayMove } from "@dnd-kit/sortable";  
import type { Draft } from "@reduxjs/toolkit";  
import { createAction, createReducer } from "@reduxjs/toolkit";  
import { t } from "ttag";  
  
import {  
 ...

REQ-2141: import { TEST\_DASHBOARD\_STATE } from "../components/DashboardTabs/test-utils";  
  
import { getIdFromSlug, moveTab, tabsReducer } from "./tabs";  
  
/\*\*  
 \* It's preferred to write tests in...

REQ-2142: import { createAction } from "@reduxjs/toolkit";  
  
import type { DisplayTheme } from "metabase/public/lib/types";  
  
export const SET\_DISPLAY\_THEME = "metabase/dashboard/SET\_DISPLAY\_THEME";  
export const...

REQ-2143: import { t } from "ttag";  
import \_ from "underscore";  
  
import { getTrashUndoMessage } from "metabase/archive/utils";  
import { canonicalCollectionId } from "metabase/collections/utils";  
import {...

REQ-2144: import { SIDEBAR\_NAME } from "metabase/dashboard/constants";  
import { createAction, createThunkAction } from "metabase/lib/redux";  
import type { DashCardId } from "metabase-types/api";  
import type {  
...

REQ-2145: import { t } from "ttag";  
import \_ from "underscore";  
  
import { getIframeDomainName } from "metabase/visualizations/visualizations/IFrameViz/utils";  
import type {  
 DashCardId,  
 Dashboard,  
 ...

REQ-2146: import {  
 createMockActionDashboardCard,  
 createMockDashboard,  
 createMockDashboardCard,  
 createMockHeadingDashboardCard,  
 createMockLinkDashboardCard,  
 createMockTextDashboardCard,  
} from...

REQ-2147: import { trackSchemaEvent, trackSimpleEvent } from "metabase/lib/analytics";  
import type {  
 DashboardId,  
 DashboardWidth,  
 VisualizationDisplay,  
} from "metabase-types/api";  
  
import type {...

REQ-2148: export \* from "./ActionSidebar";

REQ-2149: export \* from "./AddCardSidebar";

REQ-2150: export \* from "./AddFilterParameterMenu";

REQ-2151: export \* from "./SidebarItem";

REQ-2152: import { t } from "ttag";  
  
import { hasActionsMenu } from "metabase/lib/click-behavior";  
import { useSelector } from "metabase/lib/redux";  
import { getApplicationName } from...

REQ-2153: import type { IconName } from "metabase/ui";  
import { getColumnSettings } from "metabase-lib/v1/queries/utils/column-key";  
import type {  
 ClickBehaviorType,  
 DashboardCard,  
 DatasetColumn,  
} from...

REQ-2154: export \* from "./CollapsibleDashboardParameterList";

REQ-2155: export \* from "./DashCardActionButton";

REQ-2156: import type { MantineColor } from "@mantine/core/lib/core";  
import type { ReactNode } from "react";  
  
import type { DashboardContextProps } from "metabase/dashboard/context";  
import type { IconName }...

REQ-2157: import { PLUGIN\_FEATURE\_LEVEL\_PERMISSIONS } from "metabase/plugins";  
import \* as Lib from "metabase-lib";  
import type Question from "metabase-lib/v1/Question";  
import type { Dataset } from...

REQ-2158: export { DisabledNativeCardHelpText } from "./DisabledNativeCardHelpText";

REQ-2159: import { screen } from "\_\_support\_\_/ui";  
import { createMockParameter } from "metabase-types/api/mocks";  
  
import { setup } from "./setup";  
  
describe("DashCardParameterMapper >...

REQ-2160: import { screen } from "\_\_support\_\_/ui";  
  
import type { SetupOpts } from "./setup";  
import { setup as baseSetup } from "./setup";  
  
function setup(opts: SetupOpts) {  
 baseSetup({...

REQ-2161: import { screen } from "\_\_support\_\_/ui";  
  
import type { SetupOpts } from "./setup";  
import { setup as baseSetup } from "./setup";  
  
function setup(opts: SetupOpts) {  
 baseSetup({  
 ...

REQ-2162: import { isJWT } from "metabase/lib/utils";  
import { isUuid } from "metabase/lib/uuid";  
import type { DashboardCard } from "metabase-types/api";  
  
export const getDashcardTokenId = (dashcard:...

REQ-2163: import type { Card, DashboardCard } from "metabase-types/api";  
  
export type CardSlownessStatus = "usually-fast" | "usually-slow" | boolean;  
  
export type NavigateToNewCardFromDashboardOpts = {  
 ...

REQ-2164: import { getVirtualCardType } from "metabase/dashboard/utils";  
import type {  
 BaseDashboardCard,  
 Series,  
 VisualizerColumnValueSource,  
 VisualizerDataSourceId,  
 VisualizerVizDefinition,  
} from...

REQ-2165: import type { VisualizerVizDefinition } from "metabase-types/api";  
import {  
 createMockCard,  
 createMockColumn,  
 createMockDataset,  
 createMockDatasetData,  
 createMockSeries,  
 ...

REQ-2166: export { Grid } from "./Grid";  
export { ParametersList } from "./ParametersList";

REQ-2167: import { useCallback } from "react";  
  
import { setDashboardAttributes } from "metabase/dashboard/actions";  
import { getDashboardComplete } from "metabase/dashboard/selectors";  
import { useDispatch,...

REQ-2168: import { DASHBOARD\_ACTION } from "./dashboard-action-keys";  
  
// Buttons visible in public dashboard  
export const DASHBOARD\_DISPLAY\_ACTIONS = [  
 DASHBOARD\_ACTION.REFRESH\_WIDGET,  
 ...

REQ-2169: import type { ComponentType } from "react";  
  
import type { DashboardContextReturned } from "metabase/dashboard/context";  
import type { Collection, Dashboard } from "metabase-types/api";  
  
import type...

REQ-2170: export \* from "./AddFilterParameterButton";

REQ-2171: export \* from "./AddSectionButton";

REQ-2172: export \* from "./DashboardHeader";

REQ-2173: import { screen } from "\_\_support\_\_/ui";  
import {  
 createMockCollection,  
 createMockDashboard,  
} from "metabase-types/api/mocks";  
  
import { setup } from "./setup";  
  
const setupEnterprise = (opts:...

REQ-2174: export { DashboardEntityIdCard } from "./DashboardEntityIdCard";

REQ-2175: export \* from "./DashboardInfoSidebar";

REQ-2176: import userEvent from "@testing-library/user-event";  
  
import { screen } from "\_\_support\_\_/ui";  
import type { Dashboard } from "metabase-types/api";  
import {  
 createMockCollection,  
 ...

REQ-2177: import { screen } from "\_\_support\_\_/ui";  
import type { Dashboard } from "metabase-types/api";  
import { createMockDashboard } from "metabase-types/api/mocks";  
  
import type { SetupOpts } from...

REQ-2178: import { screen } from "\_\_support\_\_/ui";  
import type { Dashboard } from "metabase-types/api";  
import {  
 createMockDashboard,  
 createMockSettings,  
 createMockTokenFeatures,  
} from...

REQ-2179: import type { Location } from "history";  
  
import { deserializeCard, parseHash } from "metabase/query\_builder/actions";  
  
export const isNavigatingToCreateADashboardQuestion = (  
 nextLocation?:...

REQ-2180: export \* from "./DashboardParameterList";

REQ-2181: export \* from "./DashboardParameterPanel";

REQ-2182: export \* from "./DashboardSettingsSidebar";

REQ-2183: import userEvent from "@testing-library/user-event";  
  
import { screen } from "\_\_support\_\_/ui";  
  
import { setup } from "./setup";  
  
jest.mock("metabase/dashboard/constants", () => ({  
 ...

REQ-2184: import { screen } from "\_\_support\_\_/ui";  
  
import { setupEnterprise } from "./setup";  
  
describe("DashboardSettingsSidebar > enterprise", () => {  
 it("should render the component", async () => {  
 ...

REQ-2185: import userEvent from "@testing-library/user-event";  
  
import { screen } from "\_\_support\_\_/ui";  
import {  
 createMockCollection,  
 createMockDashboard,  
} from "metabase-types/api/mocks";  
  
import {...

REQ-2186: export \* from "./DashboardTabs";  
export \* from "./use-dashboard-tabs";

REQ-2187: import { getDefaultTab } from "metabase/dashboard/actions";  
import { INITIAL\_DASHBOARD\_STATE } from "metabase/dashboard/constants";  
import {  
 createMockCard,  
 createMockDashboardCard,  
} from...

REQ-2188: import type { UniqueIdentifier } from "@dnd-kit/core";  
import { t } from "ttag";  
  
import { trackTabDuplicated } from "metabase/dashboard/analytics";  
import { useDashboardContext } from...

REQ-2189: export \* from "./DashboardTitle";

REQ-2190: export \* from "./ExtraEditButtonsMenu";

REQ-2191: import { push } from "react-router-redux";  
  
import { getDashboard } from "metabase/dashboard/selectors";  
import { createThunkAction } from "metabase/lib/redux";  
import \* as Urls from...

REQ-2192: export \* from "./QuestionPicker";

REQ-2193: export \* from "./RefreshOption";

REQ-2194: export \* from "./RefreshWidget";

REQ-2195: import \_ from "underscore";  
  
import { getMobileHeight } from "metabase/visualizations/shared/utils/sizes";  
import type { BaseDashboardCard } from "metabase-types/api";  
  
function...

REQ-2196: import { MOBILE\_DEFAULT\_CARD\_HEIGHT } from "metabase/visualizations/shared/utils/sizes";  
import type { VisualizationDisplay } from "metabase-types/api";  
import {  
 createMockCard,  
 ...

REQ-2197: import type {  
 DashboardSidebarName,  
 DashboardState,  
} from "metabase-types/store";  
  
import type { EmbedDisplayParams } from "./types";  
  
export const DASHBOARD\_DESCRIPTION\_MAX\_LENGTH =...

REQ-2198: export \* from "./AutomaticDashboardApp";

REQ-2199: import type { WithRouterProps } from "react-router";  
  
import { useDashboardContext } from "metabase/dashboard/context";  
import { useLocationSync } from "metabase/dashboard/hooks";  
import type {...

REQ-2200: import { useCallback, useEffect } from "react";  
import { t } from "ttag";  
  
import { useLoadingTimer } from "metabase/common/hooks/use-loading-timer";  
import { useUniqueId } from...

REQ-2201: export \* from "./DashboardSharingEmbeddingModal";

REQ-2202: import type { ConnectedProps } from "react-redux";  
import { push } from "react-router-redux";  
  
import { deletePermanently } from "metabase/archive/actions";  
import {  
 addCardToDashboard,  
 ...

REQ-2203: export \* from "./context";

REQ-2204: import \_ from "underscore";  
  
import { DEFAULT\_CARD\_SIZE } from "metabase/lib/dashboard\_grid";  
import { getVisualizationRaw } from "metabase/visualizations";  
import type {  
 BaseDashboardCard,  
 ...

REQ-2205: export \* from "./use-click-behavior-data";  
export \* from "./use-dashboard-fullscreen";  
export \* from "./use-dashboard-refresh-period";  
export \* from "./use-dashboard-theme";  
export \* from...

REQ-2206: import type { LocationDescriptorObject } from "history";  
import { useCallback, useMemo } from "react";  
import { replace } from "react-router-redux";  
  
import { parseHashOptions, stringifyHashOptions }...

REQ-2207: import { useMemo } from "react";  
import \_ from "underscore";  
  
import {  
 getDashCardById,  
 getDashboardComplete,  
 getParameterValuesBySlugMap,  
 getParameters,  
} from...

REQ-2208: import { useFullscreen } from "@mantine/hooks";  
import { useCallback, useEffect, useState } from "react";  
  
import type { DashboardFullscreenControls } from "../types";  
  
export const...

REQ-2209: import { useCallback, useRef, useState } from "react";  
import { useUnmount } from "react-use";  
  
import type { DashboardRefreshPeriodControls } from "../types";  
  
import { useInterval } from...

REQ-2210: import { useCallback, useEffect } from "react";  
  
import { setDisplayTheme } from "metabase/dashboard/actions";  
import { getDisplayTheme } from "metabase/dashboard/selectors";  
import { useDispatch,...

REQ-2211: import { useCallback } from "react";  
  
import { useSetDashboardAttributeHandler } from "../components/Dashboard/use-set-dashboard-attribute";  
import { useDashboardContext } from "../context";  
  
export...

REQ-2212: import type { Location } from "history";  
import { useEffect, useMemo } from "react";  
import type { InjectedRouter } from "react-router";  
import { push, replace } from "react-router-redux";  
import {...

REQ-2213: import { useCallback, useEffect, useRef, useState } from "react";  
  
/\*\*  
 \* it's a copy of mantine's useInterval, but with memoization of the callbacks  
 \*/  
export function useInterval(fn: () => void,...

REQ-2214: import { type RefObject, useEffect, useState } from "react";  
  
export function useIsParameterPanelSticky({  
 parameterPanelRef,  
 disabled = false,  
}: {  
 parameterPanelRef: RefObject<HTMLElement>;  
 ...

REQ-2215: import type { Location } from "history";  
import { useEffect, useMemo } from "react";  
import { replace } from "react-router-redux";  
import { usePrevious } from "react-use";  
import { omit } from...

REQ-2216: import type { Query } from "history";  
import { useCallback } from "react";  
  
import {  
 fetchDashboard,  
 fetchDashboardCardData,  
} from "metabase/dashboard/actions";  
import { useDispatch } from...

REQ-2217: import { useRegisterMetabotContextProvider } from "metabase/metabot";  
  
import { getDashboard } from "../selectors";  
  
export const useRegisterDashboardMetabotContext = () => {  
 ...

REQ-2218: import { useCallback, useEffect, useRef, useState } from "react";  
  
import { useDashboardContext } from "metabase/dashboard/context";  
import resizeObserver from "metabase/lib/resize-observer";  
  
type...

REQ-2219: import { createReducer } from "@reduxjs/toolkit";  
import { assocIn, dissocIn } from "icepick";  
import { omit } from "underscore";  
  
import {  
 createDashboardPublicLink,  
 deleteDashboardPublicLink,  
 ...

REQ-2220: import { assoc, assocIn, chain, dissoc, merge, updateIn } from "icepick";  
import reduceReducers from "reduce-reducers";  
import \_ from "underscore";  
  
import Actions from...

REQ-2221: import { createMockDashboard } from "metabase-types/api/mocks";  
  
import {  
 ADD\_DASHCARD\_IDS\_TO\_LOADING\_QUEUE,  
 CLOSE\_SIDEBAR,  
 FETCH\_DASHBOARD\_CARD\_DATA,  
 INITIALIZE,  
 REMOVE\_PARAMETER,  
 ...

REQ-2222: import { t } from "ttag";  
  
import { GRID\_WIDTH } from "metabase/lib/dashboard\_grid";  
import type {  
 DashboardCardLayoutAttrs,  
 VirtualCard,  
 VirtualDashboardCard,  
} from...

REQ-2223: import { createSelector } from "@reduxjs/toolkit";  
import { createCachedSelector } from "re-reselect";  
import \_ from "underscore";  
  
import { LOAD\_COMPLETE\_FAVICON } from...

REQ-2224: import { chain } from "icepick";  
  
import { createMockEntitiesState } from "\_\_support\_\_/store";  
import {  
 getClickBehaviorSidebarDashcard,  
 getDashboardComplete,  
 getEditingParameterId,  
 ...

REQ-2225: import type { EmbedResourceDownloadOptions } from "metabase/public/lib/types";  
  
export type DashboardFullscreenControls = {  
 isFullscreen: boolean;  
 onFullscreenChange: (  
 newIsFullscreen:...

REQ-2226: import type { DashboardNightModeControls } from "metabase/dashboard/types/display-options";  
import type {  
 DisplayTheme,  
 EmbedResourceDownloadOptions,  
} from "metabase/public/lib/types";  
import...

REQ-2227: import type { Dashboard } from "metabase-types/api";  
  
export type SuccessfulFetchDashboardResult = {  
 payload: { dashboard: Dashboard };  
};  
export type FailedFetchDashboardResult = { error: unknown;...

REQ-2228: import type {  
 EmbeddingAdditionalHashOptions,  
 EmbeddingDisplayOptions,  
} from "metabase/public/lib/types";  
  
/\*\*  
 \* This is a type that controls some dashboard states.  
 \*/  
export type...

REQ-2229: export \* from "./display-options";  
export \* from "./hash-options";  
export \* from "./embed-display-options";  
export \* from "./fetch-dashboard-result";

REQ-2230: import type { Location } from "history";  
import \_ from "underscore";  
  
import { SERVER\_ERROR\_TYPES } from "metabase/lib/errors";  
import { isJWT } from "metabase/lib/utils";  
import { isUuid } from...

REQ-2231: import type { Location } from "history";  
  
import {  
 canResetFilter,  
 createTabSlug,  
 fetchDataOrError,  
 findDashCardForInlineParameter,  
 getCurrentTabDashboardCards,  
 getDashcardResultsError,  
 ...

REQ-2232: export const ROW\_HEIGHT = 36;  
export const HEADER\_HEIGHT = 36;  
export const ADD\_COLUMN\_BUTTON\_WIDTH = 36;  
export const MIN\_COLUMN\_WIDTH = 60;  
export const ROW\_ID\_COLUMN\_ID = `\0\_INDEX`;  
export const...

REQ-2233: import type { MaybeVirtualRow, VirtualRow } from "./types";  
  
export const isVirtualRow = <TData>(  
 maybeVirtualRow: MaybeVirtualRow<TData>,  
): maybeVirtualRow is VirtualRow<TData> => {  
 return...

REQ-2234: export { useDataGridTheme, DataGridThemeProvider } from "./use-table-theme";

REQ-2235: export \* from "./components/DataGrid/DataGrid";  
export \* from "./components/BaseCell/BaseCell";  
export \* from "./components/HeaderCell/HeaderCell";  
export \* from...

REQ-2236: import type {  
 Cell,  
 CellContext,  
 ColumnDefTemplate,  
 ColumnPinningState,  
 ColumnSizingState,  
 HeaderContext,  
 OnChangeFn,  
 Row,  
 RowData,  
 RowSelectionOptions,  
 RowSelectionState,  
 ...

REQ-2237: import type { ColumnSizingState } from "@tanstack/react-table";  
  
export const pickRowsToMeasure = <TData, TValue>(  
 data: TData[],  
 accessorFn: (row: TData) => TValue,  
 count = 10,  
) => {  
 const...

REQ-2238: import { getTruncatedColumnSizing, pickRowsToMeasure } from "./column-sizing";  
  
describe("pickRowsToMeasure", () => {  
 const createData = (values: (string | null | undefined)[]) =>  
 ...

REQ-2239: import type { PlainCellFormatter } from "../types";  
  
/\*\*  
 \* Formats cell value for copying, applying formatters when available.  
 \*  
 \* Null value handling:  
 \* - Raw copy: null/undefined -> "null"  
 \* -...

REQ-2240: import type { ColumnSizingState } from "@tanstack/react-table";  
  
/\*\*  
 \* Proportionally expands column widths to fill available space while preserving fixed-width columns.  
 \* Only expands columns when...

REQ-2241: import { maybeExpandColumnWidths } from "./maybe-expand-column-widths";  
  
describe("maybeExpandColumnWidths", () => {  
 it("returns the original map when minGridWidth is undefined, 0, or negative", ()...

REQ-2242: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2243: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2244: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2245: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2246: export { DatabaseConnectionSectionField } from "./DatabaseConnectionSectionField";

REQ-2247: import { trackSimpleEvent } from "metabase/lib/analytics";  
  
export function connectionStringParsedSuccess(  
 location: "admin" | "setup" | "embedding\_setup",  
) {  
 trackSimpleEvent({  
 event:...

REQ-2248: import { P, match } from "ts-pattern";  
  
import type { EngineKey } from "metabase-types/api/settings";  
  
import type { RegexFields } from "./parse-connection-regex";  
  
function...

REQ-2249: import type { EngineKey } from "metabase-types/api/settings";  
  
type Placeholder = string;  
  
export const enginesConfig: Record<EngineKey, Placeholder> = {  
 athena:...

REQ-2250: export { DatabaseConnectionStringField } from "./DatabaseConnectionStringField";

REQ-2251: import type { EngineKey } from "metabase-types/api";  
  
export interface RegexFields {  
 host?: string;  
 port?: string;  
 database?: string;  
 catalog?: string;  
 schema?: string;  
 username?:...

REQ-2252: import { enginesConfig } from "./engines-config";  
import { parseConnectionUriRegex } from "./parse-connection-regex";  
  
describe("parseConnectionUriRegex - Amazon Athena", () => {  
 it("should parse a...

REQ-2253: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2254: export { DatabaseEngineField } from "./DatabaseEngineField";

REQ-2255: export { DatabaseEngineList } from "./DatabaseEngineList";

REQ-2256: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2257: export \* from "./DatabaseForm";

REQ-2258: export \* from "./DatabaseHelpCard";

REQ-2259: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2260: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2261: export { DatabaseNameField } from "./DatabaseNameField";

REQ-2262: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2263: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2264: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2265: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2266: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2267: import { getSetting } from "metabase/selectors/settings";  
import type { State } from "metabase-types/store";  
  
export const getEngines = (state: State) => {  
 return getSetting(state,...

REQ-2268: import type { ComponentType, ReactNode } from "react";  
  
import type { EngineFieldOption, EngineFieldType } from "metabase-types/api";  
  
export interface EngineOption {  
 name: string;  
 value:...

REQ-2269: import type { Engine } from "metabase-types/api";  
  
import { ELEVATED\_ENGINES, ENGINE\_LOGO } from "../constants";  
import type { EngineOption } from "../types";  
  
export const getEngineOptions = (  
 ...

REQ-2270: import type { TestContext } from "yup";  
import \* as Yup from "yup";  
  
import \* as Errors from "metabase/lib/errors";  
import type { DatabaseData, Engine, EngineField } from...

REQ-2271: Make sure this gets recognized as an ES module

REQ-2272: if (  
 // eslint-disable-next-line no-undef  
 process.env.ENABLE\_CLJS\_HOT\_RELOAD === "true" ||  
 // eslint-disable-next-line no-undef  
 process.env.ENABLE\_CLJS\_DEV\_TOOLS === "true"  
) {  
 ...

REQ-2273: export { QuestionPublicLinkPopover } from "./QuestionPublicLinkPopover";  
export { DashboardPublicLinkPopover } from "./DashboardPublicLinkPopover";

REQ-2274: import type { exportFormats } from "metabase/lib/urls";  
  
export type ExportFormatType = (typeof exportFormats)[number] | null;

REQ-2275: export \* from "./DashboardSharingMenu";  
export \* from "./QuestionSharingMenu";

REQ-2276: export type EmbedModalType = "question-embed" | "dashboard-embed";  
export type PublicLinkModalType =  
 | "question-public-link"  
 | "dashboard-public-link";  
  
export type QuestionSharingModalType =  
 ...

REQ-2277: import type { SettingKey, TokenFeature } from "metabase-types/api";  
  
export const EMBEDDING\_SDK\_PORTAL\_ROOT\_ELEMENT\_ID = "metabase-sdk-portal-root";  
  
type InternalSdkConfig = {  
 isEmbeddingSdk:...

REQ-2278: import \* as React from "react";  
import \* as ReactJSXRuntime from "react/jsx-runtime";  
import \* as ReactDOM from "react-dom";  
import \* as ReactDOMClient from "react-dom/client";  
import \* as...

REQ-2279: \*

REQ-2280: const CONFLICTING\_CLJS\_GLOBALS = ["cljs", "malli", "metabase"];  
  
// To properly work with CLJS runtime that comes with Embedding SDK bundle  
export function renameConflictingCljsGlobals() {  
 for...

REQ-2281: import type { CSSProperties } from "react";  
  
import type { ColorName } from "metabase/lib/colors/types";  
  
import type { DeepPartial } from "../types/utils";  
  
import type { MetabaseFontFamily } from...

REQ-2282: import type { FlattenObjectKeys } from "../types/utils";  
  
import type { MetabaseComponentTheme } from "./MetabaseTheme";  
  
type MetabaseComponentThemeKey =...

REQ-2283: import { merge } from "icepick";  
  
import { OVERLAY\_Z\_INDEX } from "metabase/css/core/overlays/constants";  
import { EMBEDDING\_SDK\_PORTAL\_ROOT\_ELEMENT\_ID } from "metabase/embedding-sdk/config";  
import...

REQ-2284: import type { MetabaseTheme } from "./MetabaseTheme";  
  
export const defineMetabaseTheme = (theme: MetabaseTheme): MetabaseTheme =>  
 theme;

REQ-2285: import type { DynamicCssVarConfig } from "../types/private/css-variables";  
  
/\*\*  
 \* These CSS variables are dynamically generated based on the theme.  
 \*/  
export const DYNAMIC\_CSS\_VARIABLES:...

REQ-2286: eslint-disable-next-line no-restricted-imports

REQ-2287: import type {  
 MetabaseColor,  
 MetabaseColors,  
} from "metabase/embedding-sdk/theme";  
import { colors } from "metabase/lib/colors";  
import type { ColorName, ColorPalette } from...

REQ-2288: export type MetabaseFontFamily =  
 | "Roboto"  
 | "Merriweather"  
 | "Open Sans"  
 | "Lato"  
 | "Noto Sans"  
 | "Roboto Slab"  
 | "Source Sans Pro"  
 | "Raleway"  
 | "Slabo 27px"  
 | "PT Sans"  
 |...

REQ-2289: import type { ChartColor } from "metabase/embedding-sdk/theme";  
  
/\*\*  
 \* Map the input chart colors from the theme settings to the  
 \* color names we use in our charts.  
 \*  
 \* @param chartColors chart...

REQ-2290: export \* from "./default-component-theme";  
// eslint-disable-next-line no-literal-metabase-strings -- file name  
export \* from "./MetabaseTheme";  
  
export \* from "./define-metabase-theme";

REQ-2291: import type { MetabaseComponentTheme, MetabaseTheme } from ".";  
  
/\*\*  
 \* Mantine theme options specific to React embedding.  
 \*/  
export type EmbeddingThemeOptions = MetabaseComponentTheme &  
 ...

REQ-2292: import type { CardType, TableId } from "metabase-types/api";  
  
export interface DataSourceSelectorProps {  
 isInitiallyOpen: boolean;  
 /\*\* Type of the the query's first stage \*/  
 querySourceType:  
 ...

REQ-2293: export type ParameterValues = Record<  
 string,  
 string | string[] | undefined | null  
>;

REQ-2294: import type { IconName as InternalIconName } from "metabase/ui";  
  
/\*\*  
 \* Inline wrapper to properly display the `IconName` type without referencing the `internal` type  
 \*  
 \* @hidden  
 \* @inline  
...

REQ-2295: import type { ReactNode } from "react";  
  
import type { DashCardMenuItem } from "metabase/dashboard/components/DashCard/DashCardMenu/dashcard-menu";  
import type { ClickAction, ClickObject } from...

REQ-2296: import type { SemanticColorKey } from "metabase/embedding-sdk/theme/embedding-color-palette";  
import type { ColorName } from "metabase/lib/colors/types";  
  
export type SourceColorKey = ColorName |...

REQ-2297: export interface MetabaseQuestion {  
 id: number;  
 name: string;  
 description: string | null;  
 entityId: string;  
  
 isSavedQuestion: boolean;  
}

REQ-2298: \*

REQ-2299: import { updateIn } from "icepick";  
import { t } from "ttag";  
  
import {  
 actionApi,  
 useGetActionQuery,  
 useListActionsQuery,  
} from "metabase/api";  
import {  
 createEntity,  
 ...

REQ-2300: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2301: import { createSelector } from "@reduxjs/toolkit";  
import { assoc, dissoc, getIn, updateIn } from "icepick";  
import { t } from "ttag";  
import \_ from "underscore";  
  
import { bookmarkApi,...

REQ-2302: import { createSelector } from "@reduxjs/toolkit";  
import { t } from "ttag";  
import \_ from "underscore";  
  
import {  
 collectionApi,  
 skipToken,  
 useGetCollectionQuery,  
 useListCollectionsQuery,  
 ...

REQ-2303: import { t } from "ttag";  
  
export const DEFAULT\_COLLECTION\_COLOR\_ALIAS = "brand";  
  
export const ROOT\_COLLECTION = {  
 id: "root" as const,  
 get name() {  
 return t`Our analytics`;  
 },  
 location:...

REQ-2304: import { t } from "ttag";  
import \_ from "underscore";  
  
import {  
 PERSONAL\_COLLECTION,  
 PERSONAL\_COLLECTIONS,  
 ROOT\_COLLECTION,  
} from "./constants";  
  
// given list of collections with { id, name,...

REQ-2305: import { createSelector } from "@reduxjs/toolkit";  
import type { Location } from "history";  
  
import {  
 canonicalCollectionId,  
 isRootTrashCollection,  
} from "metabase/collections/utils";  
import \*...

REQ-2306: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2307: import { collections as Collections } from "metabase/entities";  
import { ROOT\_COLLECTION } from "metabase/entities/collections/constants";  
  
describe("Collection selectors", () => {  
 const...

REQ-2308: import {  
 isRootCollection,  
 isRootPersonalCollection,  
 isRootTrashCollection,  
} from "metabase/collections/utils";  
import { color } from "metabase/lib/colors";  
import { PLUGIN\_COLLECTIONS } from...

REQ-2309: import { setupEnterpriseTest } from "\_\_support\_\_/enterprise";  
import { createMockCollection } from "metabase-types/api/mocks";  
  
import { PERSONAL\_COLLECTIONS } from "./constants";  
import {  
 ...

REQ-2310: import "mutationobserver-shim";  
  
import { renderWithProviders } from "\_\_support\_\_/ui";  
import { EntityListLoader } from "metabase/entities/containers/rtk-query";  
import { Api } from...

REQ-2311: eslint-disable react/prop-types

REQ-2312: export \* from "./EntityListLoader";  
export \* from "./EntityObjectLoader";  
export \* from "./types";

REQ-2313: import type { BaseQueryFn, QueryDefinition } from "@reduxjs/toolkit/query";  
  
import type { TagType } from "metabase/api/tags";  
import type { IconName } from "metabase/ui";  
import type { Collection }...

REQ-2314: export \* from "./entities";

REQ-2315: import type {  
 QueryActionCreatorResult,  
 QueryArgFrom,  
 QueryDefinition,  
 QueryStatus,  
 QuerySubState,  
 ResultTypeFrom,  
 SkipToken,  
 SubscriptionOptions,  
 TSHelpersId,  
 TSHelpersNoInfer,  
 ...

REQ-2316: import { useCallback, useMemo, useState } from "react";  
  
import type { EntityQuery } from "./types";  
  
/\*\*  
 \* @deprecated exists for backwards compatibility  
 \*/  
export const usePaginatedQuery = (  
 ...

REQ-2317: import { t } from "ttag";  
  
import { automagicDashboardsApi } from "metabase/api/automagic-dashboards";  
import {  
 dashboardApi,  
 useGetDashboardQuery,  
 useListDashboardsQuery,  
} from...

REQ-2318: import { createSelector } from "@reduxjs/toolkit";  
import { normalize } from "normalizr";  
import \_ from "underscore";  
  
import {  
 databaseApi,  
 useGetDatabaseMetadataQuery,  
 useGetDatabaseQuery,  
 ...

REQ-2319: import { assocIn, updateIn } from "icepick";  
import { normalize } from "normalizr";  
import { useMemo } from "react";  
import { t } from "ttag";  
  
import {  
 fieldApi,  
 skipToken,  
 useGetFieldQuery,  
 ...

REQ-2320: import { assocIn } from "icepick";  
  
import { CLEAR\_MEMBERSHIPS } from "metabase/admin/people/events";  
import {  
 permissionApi,  
 useGetPermissionsGroupQuery,  
 useListPermissionsGroupsQuery,  
} from...

REQ-2321: export { default as actions } from "./actions";  
export { default as collections } from "./collections";  
export { default as snippetCollections } from "./snippet-collections";  
export { default as...

REQ-2322: export \* from "./indexed-entities";

REQ-2323: \*

REQ-2324: import { dissocIn } from "icepick";  
import \_ from "underscore";  
  
import {  
 createModelIndex,  
 deleteModelIndex,  
 listModelIndexes,  
} from "metabase/api";  
import type Question from...

REQ-2325: import fetchMock from "fetch-mock";  
  
import { getStore } from "\_\_support\_\_/entities-store";  
import { setupModelIndexEndpoints } from "\_\_support\_\_/server-mocks";  
import { Api } from...

REQ-2326: import \_ from "underscore";  
  
import type Question from "metabase-lib/v1/Question";  
import type FieldEntity from "metabase-lib/v1/metadata/Field";  
import {  
 isBoolean,  
 isInteger,  
 isPK,  
 ...

REQ-2327: import Question from "metabase-lib/v1/Question";  
import Field from "metabase-lib/v1/metadata/Field";  
import type { Field as FieldAPI } from "metabase-types/api";  
import { createMockCard,...

REQ-2328: import { createSelector } from "@reduxjs/toolkit";  
  
import {  
 cardApi,  
 persistApi,  
 skipToken,  
 useGetPersistedInfoByCardQuery,  
 useGetPersistedInfoQuery,  
 useListPersistedInfoQuery,  
} from...

REQ-2329: import { t } from "ttag";  
  
import {  
 subscriptionApi,  
 useGetSubscriptionQuery,  
 useListSubscriptionsQuery,  
} from "metabase/api";  
import { getCollectionType } from...

REQ-2330: import { updateIn } from "icepick";  
import { t } from "ttag";  
  
import {  
 cardApi,  
 datasetApi,  
 useGetCardQuery,  
 useListCardsQuery,  
} from "metabase/api";  
import {  
 canonicalCollectionId,  
 ...

REQ-2331: import { revisionApi, useListRevisionsQuery } from "metabase/api";  
import { createEntity, entityCompatibleQuery } from "metabase/lib/entities";  
  
import Dashboards from "./dashboards";  
import...

REQ-2332: import { assocIn, updateIn } from "icepick";  
import { useMemo } from "react";  
  
import {  
 databaseApi,  
 skipToken,  
 useListDatabaseSchemaTablesQuery,  
 useListDatabaseSchemasQuery,  
 ...

REQ-2333: import fetchMock from "fetch-mock";  
  
import { getStore } from "\_\_support\_\_/entities-store";  
import { Api } from "metabase/api";  
import Questions from "metabase/entities/questions";  
import Schemas...

REQ-2334: import { useMemo } from "react";  
  
import {  
 cardApi,  
 collectionApi,  
 searchApi,  
 skipToken,  
 useListCollectionItemsQuery,  
 useSearchQuery,  
} from "metabase/api";  
import { canonicalCollectionId...

REQ-2335: import {  
 segmentApi,  
 useGetSegmentQuery,  
 useListSegmentsQuery,  
} from "metabase/api";  
import { color } from "metabase/lib/colors";  
import { createEntity, entityCompatibleQuery } from...

REQ-2336: import { createSelector } from "@reduxjs/toolkit";  
import { t } from "ttag";  
import \_ from "underscore";  
  
import { skipToken, useGetCollectionQuery } from "metabase/api";  
import {...

REQ-2337: import {  
 snippetApi,  
 useGetSnippetQuery,  
 useListSnippetsQuery,  
} from "metabase/api";  
import { createEntity, entityCompatibleQuery } from "metabase/lib/entities";  
  
/\*\*  
 \* @deprecated use...

REQ-2338: import { createSelector } from "@reduxjs/toolkit";  
import { updateIn } from "icepick";  
import { useEffect, useMemo } from "react";  
import { match } from "ts-pattern";  
import { t } from "ttag";  
import...

REQ-2339: import Questions from "metabase/entities/questions";  
import Tables from "metabase/entities/tables";  
import { convertSavedQuestionToVirtualTable } from...

REQ-2340: import { t } from "ttag";  
  
import { timelineEventApi, useGetTimelineEventQuery } from "metabase/api";  
import {  
 createEntity,  
 entityCompatibleQuery,  
 undo,  
} from "metabase/lib/entities";  
import...

REQ-2341: import { updateIn } from "icepick";  
import { t } from "ttag";  
import \_ from "underscore";  
  
import {  
 skipToken,  
 timelineApi,  
 timelineEventApi,  
 useGetTimelineQuery,  
 ...

REQ-2342: const tryOrDefault = <T>(fn: () => T, defaultValue: T): T => {  
 try {  
 return fn();  
 } catch (e) {  
 console.warn(  
 "Error while trying to get env",  
 e,  
 `returning default:...

REQ-2343: export \* from "./Form";

REQ-2344: export \* from "./FormCheckbox";

REQ-2345: export \* from "./FormCheckboxGroup";

REQ-2346: export \* from "./FormChipGroup";

REQ-2347: export \* from "./FormDateInput";

REQ-2348: export \* from "./FormErrorMessage";

REQ-2349: export \* from "./FormGroupWidget";

REQ-2350: export \* from "./FormGroupsWidget";

REQ-2351: export \* from "./FormMessage";

REQ-2352: export \* from "./FormNumberInput";

REQ-2353: export \* from "./FormObserver";

REQ-2354: export \* from "./FormProvider";

REQ-2355: export \* from "./FormRadioGroup";

REQ-2356: export \* from "./FormSecretKey";

REQ-2357: export \* from "./FormSection";

REQ-2358: export \* from "./FormSelect";

REQ-2359: export \* from "./FormSubmitButton";

REQ-2360: export \* from "./FormSwitch";

REQ-2361: export \* from "./FormTextInput";

REQ-2362: export \* from "./FormTextarea";

REQ-2363: export \* from "./Form";  
export \* from "./FormCheckbox";  
export \* from "./FormCheckboxGroup";  
export \* from "./FormChipGroup";  
export \* from "./FormDateInput";  
export \* from...

REQ-2364: export \* from "./FormContext";

REQ-2365: export \* from "./FormContext";

REQ-2366: export \* from "./use-form-context";  
export \* from "./use-form-error-message";  
export \* from "./use-form-submit";  
export \* from "./use-form-submit-button";  
export \* from "./use-form-validation";

REQ-2367: export \* from "./use-form-context";

REQ-2368: import { useContext } from "react";  
  
import type { IFormContext } from "../../contexts";  
import { FormContext } from "../../contexts";  
  
export const useFormContext = (): IFormContext => {  
 return...

REQ-2369: export \* from "./use-form-error-message";

REQ-2370: import { useFormikContext } from "formik";  
import { useLayoutEffect, useState } from "react";  
import { t } from "ttag";  
  
import { useFormContext } from "../use-form-context";  
  
export const...

REQ-2371: export \* from "./use-form-submit";

REQ-2372: import type { FormikErrors } from "formik";  
  
export interface FormError<T> extends FormErrorData<T> {  
 data?: string | FormErrorData<T>;  
}  
  
export interface FormErrorData<T> {  
 errors?:...

REQ-2373: import type { FormikHelpers } from "formik";  
import type { Dispatch, SetStateAction } from "react";  
import { useCallback, useState } from "react";  
  
import { getResponseErrorMessage } from...

REQ-2374: export \* from "./use-form-submit-button";

REQ-2375: import { useFormikContext } from "formik";  
import { useEffect, useLayoutEffect, useState } from "react";  
  
import type { FormStatus } from "../../contexts";  
import { useFormContext } from...

REQ-2376: export \* from "./use-form-validation";

REQ-2377: import type { FormikErrors, FormikValues } from "formik";  
import { prepareDataForValidation, yupToFormErrors } from "formik";  
import { useCallback, useMemo } from "react";  
import type { AnySchema }...

REQ-2378: export \* from "./components";  
export \* from "./contexts";  
export \* from "./hooks";  
export \* from "./utils";

REQ-2379: export \* from "./messages";

REQ-2380: import { t } from "ttag";  
  
export interface MaxLengthParams {  
 max: number;  
}  
  
export const requiredErrorMessage = () => t`Required`;  
  
export const emailErrorMessage = () => t`Must be a valid email...

REQ-2381: import { mockSettings } from "\_\_support\_\_/settings";  
  
import { getParentPath } from "./ModalRoute";  
  
const setup = (routePath, locationPath, siteURL = undefined) => {  
 if (siteURL) {  
 ...

REQ-2382: eslint-disable react/prop-types

REQ-2383: export function getDisplayName(WrappedComponent) {  
 return WrappedComponent.displayName || WrappedComponent.name || "Component";  
}

REQ-2384: export \* from "./CustomHomePageModal";

REQ-2385: eslint-disable-next-line no-restricted-imports

REQ-2386: import { updateSetting } from "metabase/admin/settings/settings";  
import { createAsyncThunk } from "metabase/lib/redux";  
import type { EmbeddingHomepageDismissReason } from...

REQ-2387: import { trackSchemaEvent } from "metabase/lib/analytics";  
import type { EmbeddingHomepageDismissReason } from "metabase-types/api";  
  
import type { EmbeddingHomepageInitialTab } from...

REQ-2388: export { EmbedHomepage } from "./EmbedHomepage";  
export { EmbedHomepageView } from "./EmbedHomepageView";  
export type { EmbedHomepageViewProps } from "./EmbedHomepageView";

REQ-2389: export type EmbeddingHomepageInitialTab = "static" | "interactive";

REQ-2390: export \* from "./HomeCaption";

REQ-2391: export \* from "./HomeCard";

REQ-2392: export \* from "./HomeContent";

REQ-2393: export \* from "./HomeGreeting";

REQ-2394: export \* from "./HomeHelpCard";

REQ-2395: export \* from "./HomeLayout";

REQ-2396: export \* from "./HomeModelCard";

REQ-2397: export \* from "./HomePage";

REQ-2398: export \* from "./HomePopularSection";

REQ-2399: export \* from "./HomeRecentSection";

REQ-2400: export \* from "./HomeXrayCard";

REQ-2401: export \* from "./HomeXraySection";

REQ-2402: import { trackSimpleEvent } from "metabase/lib/analytics";  
  
import type { ChecklistItemCTA, ChecklistItemValue } from "./types";  
  
export const trackChecklistItemExpanded = (value: ChecklistItemValue)...

REQ-2403: export \* from "./Onboarding";

REQ-2404: export type ChecklistItemValue =  
 | "database"  
 | "invite"  
 | "x-ray"  
 | "notebook"  
 | "sql"  
 | "dashboard"  
 | "subscription"  
 | "alert";  
  
export type ChecklistItemCTA = "primary" |...

REQ-2405: import { createSelector } from "@reduxjs/toolkit";  
import dayjs from "dayjs";  
  
import { getIsEmbeddingIframe } from "metabase/selectors/embed";  
import { getSetting } from...

REQ-2406: import dayjs from "dayjs";  
  
import { parseTimestamp } from "metabase/lib/time-dayjs";  
  
export const isWithinWeeks = (  
 timestamp: string,  
 weekCount: number,  
): boolean => {  
 const date =...

REQ-2407: import { PLUGIN\_CONTENT\_TRANSLATION } from "metabase/plugins";  
  
import type { ContentTranslationFunction } from "./types";  
  
/\*\* To keep the components that require content translation tidier, they...

REQ-2408: import type { ReactElement } from "react";  
  
import { setupEnterprisePlugins } from "\_\_support\_\_/enterprise";  
import { setupContentTranslationEndpoints } from...

REQ-2409: export type NonEmpty<ArrayType> = ArrayType extends (infer ItemType)[]  
 ? [ItemType, ...ItemType[]]  
 : never;  
  
export type ContentTranslationFunction = <T = string | null | undefined>(  
 msgid:...

REQ-2410: import \* as Snowplow from "@snowplow/browser-tracker";  
  
import Settings from "metabase/lib/settings";  
import { getUserId } from "metabase/selectors/user";  
  
export const trackPageView = (url) => {  
 ...

REQ-2411: import \* as Snowplow from "@snowplow/browser-tracker";  
  
import { shouldLogAnalytics } from "metabase/env";  
import Settings from "metabase/lib/settings";  
import type {  
 SchemaEvent,  
 SchemaType,  
 ...

REQ-2412: import EventEmitter from "events";  
import querystring from "querystring";  
  
import { isTest } from "metabase/env";  
import { isWithinIframe } from "metabase/lib/dom";  
import { delay } from...

REQ-2413: export function moveElement<T>(array: T[], oldIndex: number, newIndex: number) {  
 const arrayCopy = [...array];  
 arrayCopy.splice(newIndex, 0, arrayCopy.splice(oldIndex, 1)[0]);  
 return...

REQ-2414: import { SessionApi } from "metabase/services";  
  
export const deleteSession = async () => {  
 try {  
 await SessionApi.delete();  
 } catch (error) {  
 if (error.status !== 404) {  
 ...

REQ-2415: import querystring from "querystring";  
  
import { safeJsonParse } from "metabase/lib/json-parse";  
  
function parseQueryStringOptions(s: string) {  
 const options: Record<string, string | string[] |...

REQ-2416: import { b64hash\_to\_utf8, utf8\_to\_b64url } from "metabase/lib/encoding";  
import { equals } from "metabase/lib/utils";  
  
export function createCard(name = null) {  
 return {  
 name: name,  
 ...

REQ-2417: import { getIn } from "icepick";  
import { msgid, ngettext, t } from "ttag";  
  
import Dashboards from "metabase/entities/dashboards";  
import Questions from "metabase/entities/questions";  
  
export...

REQ-2418: import { t } from "ttag";  
  
import { isNotNull } from "metabase/lib/types";  
import type { Collection, CollectionId } from "metabase-types/api";  
  
export const getCrumbs = (  
 collection: Collection,  
 ...

REQ-2419: import { createMockCollection } from "metabase-types/api/mocks";  
  
import { getCrumbs } from "./collections";  
  
const collectionsById = {  
 root: createMockCollection({ id: "root", name: "Our...

REQ-2420: import { getAccentColors, getPreferredColor } from "./groups";  
import { ACCENT\_COUNT } from "./palette";  
import type { ColorPalette } from "./types";  
  
export const getColorsForValues = (  
 keys:...

REQ-2421: import \_ from "underscore";  
  
import { getColorsForValues } from "./charts";  
import { color } from "./palette";  
  
describe("charts", () => {  
 it("should use accent colors for <= 8 series", () => {  
 ...

REQ-2422: eslint-disable no-color-literals  
NOTE: DO NOT ADD COLORS WITHOUT EXTREMELY GOOD REASON AND DESIGN REVIEW  
NOTE: KEEP SYNCHRONIZED...

REQ-2423: import \_ from "underscore";  
  
import { ACCENT\_COUNT, color } from "./palette";  
import type { AccentColorOptions, ColorName, ColorPalette } from "./types";  
  
export const getAccentColors = (  
 {  
 ...

REQ-2424: import { getAccentColors } from "./groups";  
import { color } from "./palette";  
  
describe("groups", () => {  
 describe("getAccentColors", () => {  
 it("should return main accent colors without gray...

REQ-2425: export \* from "./palette";  
export \* from "./colors";

REQ-2426: import Color from "color";  
  
import type { ColorGetter } from "metabase/visualizations/types";  
  
import { colors } from "./colors";  
import type { ColorPalette } from "./types";  
  
export const...

REQ-2427: import { colors } from "./colors";  
import { color } from "./palette";  
  
describe("palette", () => {  
 it("should get a color from the palette", () => {  
 expect(color("brand")).toBeDefined();  
 ...

REQ-2428: import { scaleLinear, scaleQuantile } from "d3-scale";  
  
export const getColorScale = (  
 extent: [number, number],  
 colors: string[],  
 isQuantile: boolean = false,  
) => {  
 if (isQuantile) {  
 ...

REQ-2429: import { color } from "./palette";  
import { getColorScale } from "./scales";  
  
describe("scales", () => {  
 const colors = [color("bg-white"), color("bg-black")];  
  
 it("should interpolate colors by...

REQ-2430: import type { colors } from "./colors";  
  
export type ColorPalette = Partial<Record<keyof typeof colors, string>>;  
  
export type ColorName = keyof ColorPalette;  
  
export interface AccentColorOptions {  
 ...

REQ-2431: import React from "react";  
  
export const getMajorReactVersion = () => {  
 const versionParts = React.version.split(".").map(Number);  
  
 return versionParts[0];  
};

REQ-2432: import React from "react";  
  
import { getMajorReactVersion } from "./check-version";  
  
describe("getMajorReactVersion", () => {  
 const versions = [  
 { version: "0.14.0", expected: 0 },  
 {...

REQ-2433: type HandlerType<E> = ((event: E) => void) | undefined;  
  
export const composeEventHandlers = <E>(...handlers: HandlerType<E>[]) => {  
 return function handleEvent(event: E) {  
 ...

REQ-2434: import { t } from "ttag";  
  
export const SEARCH\_DEBOUNCE\_DURATION = 300;  
  
export const DEFAULT\_SEARCH\_LIMIT = 50;  
  
// eslint-disable-next-line ttag/no-module-declaration -- see metabase#55045  
export...

REQ-2435: METABASE\_SESSION\_COOKIE is only used for e2e tests. In normal usage cookie is set automatically by login endpoints

REQ-2436: import { t } from "ttag";  
  
import type { IconName } from "metabase/ui";  
import { TYPE } from "metabase-lib/v1/types/constants";  
import type { Field } from "metabase-types/api";  
  
interface...

REQ-2437: import { isValidCronExpression } from "cron-expression-validator";  
import cronstrue from "cronstrue";  
import { t } from "ttag";  
import { memoize } from "underscore";  
  
import MetabaseSettings from...

REQ-2438: \*

REQ-2439: NOTE: If we make changes to the algorithm or default values below we should change  
[the backend version](https://github.com/metabase/metabase/blob/master/src/metabase/dashboards/autoplace.clj).  
If...

REQ-2440: import \_ from "underscore";  
  
import \* as Pivot from "cljs/metabase.pivot.js";  
import { formatValue } from "metabase/lib/formatting";  
import { makeCellBackgroundGetter } from...

REQ-2441: import dayjs from "dayjs";  
import { t } from "ttag";  
import \_ from "underscore";  
  
import type { DayOfWeekId } from "metabase-types/api";  
  
// returns 0-6 where Sunday as 0 and Saturday as 6  
// Note:...

REQ-2442: This is the copy of https://github.com/iamkun/dayjs/pull/2060 which implements parseZone  
which we rely on when using momentjs so we need this plugin to be able to migrate from momentjs  
@authors:...

REQ-2443: import dayjs from "dayjs";  
import advancedFormat from "dayjs/plugin/advancedFormat";  
import customParseFormat from "dayjs/plugin/customParseFormat";  
import dayOfYear from...

REQ-2444: declare global {  
 interface Window {  
 // Set REMOVE\_DELAYS to true in environments where we want to remove them.  
 // For example, in Storybook we want to remove delays to make Loki tests more  
...

REQ-2445: import querystring from "querystring";  
import \_ from "underscore";  
  
import { isCypressActive, isStorybookActive } from "metabase/env";  
import MetabaseSettings from "metabase/lib/settings";  
  
// IE...

REQ-2446: const EMAIL\_REGEX =  
 /^(([^<>()[\]\\.,;:\s@\"]+(\.[^<>()[\]\\.,;:\s@\"]+)\*)|(\".+\"))@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\])|(([a-zA-Z\-0-9]+\.)+[a-zA-Z]{2,}))$/;  
  
export function...

REQ-2447: import { push } from "react-router-redux";  
import \_ from "underscore";  
  
import CS from "metabase/css/core/index.css";  
import { IFRAMED\_IN\_SELF, isWithinIframe } from "metabase/lib/dom";  
import {...

REQ-2448: escaping before base64 encoding is necessary for non-ASCII characters  
https://developer.mozilla.org/en-US/docs/Web/API/WindowBase64/btoa

REQ-2449: import type { Engine } from "metabase-types/api";  
  
export function getEngineNativeType(engine?: string): "sql" | "json" {  
 switch (engine) {  
 case "mongo":  
 case "druid":  
 return "json";  
...

REQ-2450: import {  
 formatNativeQuery,  
 getEngineNativeType,  
 getNativeQueryLanguage,  
 isDeprecatedEngine,  
} from "metabase/lib/engine";  
import type { Engine } from...

REQ-2451: /\*  
 \* # Entities abstract the interface between the back-end and the front-end.  
 \*  
 \* ## Endpoint requirements for entities:  
 \*  
 \* When fetching a list, each item of the list must include an `id`...

REQ-2452: import { combineReducers, configureStore } from "@reduxjs/toolkit";  
import promise from "redux-promise";  
  
import { combineEntities, createEntity } from "metabase/lib/entities";  
import requestsReducer...

REQ-2453: import { useMemo } from "react";  
import \_ from "underscore";  
  
import { skipToken } from "metabase/api";  
import { useTranslateEntityIdQuery } from "metabase/api/entity-id";  
import { isBaseEntityID }...

REQ-2454: import type {  
 BaseEntityId,  
 CardId,  
 CollectionId,  
 DashboardId,  
} from "metabase-types/api";  
  
export type SUPPORTED\_ENTITIES = {  
 dashboard: DashboardId;  
 card: CardId;  
 collection:...

REQ-2455: export const MAX\_ERROR\_LOGS = 20;  
  
export function captureConsoleErrors() {  
 console.errorBuffer = [];  
  
 const originalError = console.error;  
  
 console.error = function (...args) {  
 if...

REQ-2456: import { MAX\_ERROR\_LOGS, captureConsoleErrors } from "./console";  
  
describe("captureConsolErrors", () => {  
 beforeEach(() => {  
 captureConsoleErrors();  
 });  
  
 afterEach(() => {  
 ...

REQ-2457: export \* from "./messages";  
export \* from "./validation";  
export \* from "./server-error-types";  
export \* from "./console";  
  
export type { GenericErrorResponse } from "./types";

REQ-2458: import type { GenericErrorResponse } from "./types";  
  
export function getResponseErrorMessage(error: unknown): string | undefined {  
 const response = error as GenericErrorResponse | undefined;  
  
 if...

REQ-2459: import { merge } from "icepick";  
  
import { getResponseErrorMessage } from "./messages";  
  
const ERROR\_DATA\_MESSAGE = {  
 data: {  
 message: "Error from error.data.message",  
 },  
};  
  
const...

REQ-2460: export const SERVER\_ERROR\_TYPES = {  
 missingPermissions: "missing-required-permissions",  
};

REQ-2461: export type GenericErrorResponse = {  
 data?:  
 | {  
 message?: string;  
 errors?: Record<string, string>;  
 }  
 | string;  
 errors?: Record<string, string>;  
 message?:...

REQ-2462: import { c, t } from "ttag";  
  
import type { LengthParams, MaxLengthParams } from "./types";  
  
export const required = () => t`required`;  
  
export const email = () => t`must be a valid email...

REQ-2463: const DEFAULT\_TIMEOUT = 7000;  
  
/\* wrapper around fetch that allows passing a timeout prop \*/  
export function fetchWithTimeout(  
 url: string,  
 options: RequestInit & { timeout?: number } = {},  
) {  
 ...

REQ-2464: import { color } from "metabase/lib/colors";  
  
export function assignUserColors(  
 userIds: string[],  
 currentUserId: string,  
 colors = [  
 color("brand"),  
 color("accent2"),  
 ...

REQ-2465: import { color } from "metabase/lib/colors";  
  
import { assignUserColors } from "./colors";  
  
describe("lib/formatting/colors", () => {  
 it("should assign colors to users when currentUserId is...

REQ-2466: import { NULL\_DISPLAY\_VALUE } from "metabase/lib/constants";  
import type { DatasetColumn } from "metabase-types/api/dataset";  
  
export function displayNameForColumn(column: DatasetColumn): string {  
 ...

REQ-2467: export const FK\_SYMBOL = "";

REQ-2468: import { t } from "ttag";  
  
import { currency } from "cljs/metabase.util.currency";  
  
export interface CurrencyInfo {  
 symbol: string;  
 name: string;  
 symbol\_native: string;  
 decimal\_digits:...

REQ-2469: import { getCurrencyStyleOptions } from "./currency";  
  
describe("getCurrencyStyleOptions", () => {  
 it("should get currency options - USD", () => {  
 const options =...

REQ-2470: import type { DatetimeUnit } from "metabase-types/api/query";  
  
export const DEFAULT\_TIME\_STYLE = "h:mm A";  
export const DEFAULT\_DATE\_STYLE = "MMMM D, YYYY";  
  
const UNITS\_WITH\_HOUR = ["default",...

REQ-2471: import type { Field } from "metabase-types/api/field";  
  
export function formatField(field: Field) {  
 if (!field) {  
 return "";  
 }  
  
 return field.dimensions?.[0]?.name || field.display\_name ||...

REQ-2472: import \* as d3 from "d3";  
  
import { decimalCount } from "metabase/visualizations/lib/numeric";  
import { isLatitude, isLongitude } from "metabase-lib/v1/types/utils/isa";  
  
import type { OptionsType }...

REQ-2473: import { isSameOrSiteUrlOrigin } from "metabase/lib/dom";  
import {  
 formatValue,  
 getUrlProtocol,  
 isDefaultLinkProtocol,  
} from "metabase/lib/formatting";  
import { isDate } from...

REQ-2474: import { NULL\_DISPLAY\_VALUE } from "../constants";  
  
export function formatNullable<T>(value: T | null | undefined) {  
 return value ?? NULL\_DISPLAY\_VALUE;  
}

REQ-2475: import { formatNumber, numberFormatterForOptions } from "./numbers";  
  
describe("formatNumber", () => {  
 it("should respect the decimals setting even when compact is true (metabase#54063)", () => {  
 ...

REQ-2476: import { roundFloat } from "./numbers";  
  
describe("roundFloat", () => {  
 it.each([  
 // Rounding defaults to 2 digits, defined in DEFAULT\_NUMBER\_OPTIONS  
 [30, undefined, 30],  
 [1.2345,...

REQ-2477: import inflection from "inflection";  
  
export function singularize(str: string, singular?: string) {  
 return inflection.singularize(str, singular);  
}  
  
export function pluralize(str: string, plural?:...

REQ-2478: import type { Moment } from "moment-timezone"; // eslint-disable-line no-restricted-imports -- deprecated usage  
import { msgid, ngettext } from "ttag";  
  
import type { TimeOnlyOptions } from...

REQ-2479: export interface TimeOnlyOptions {  
 local?: boolean;  
 time\_enabled?: "minutes" | "milliseconds" | "seconds" | null;  
 time\_format?: string;  
 time\_style?: string;  
}  
  
export interface OptionsType...

REQ-2480: export \* from "./formatting/colors";  
export \* from "./formatting/column";  
export \* from "./formatting/currency";  
export \* from "./formatting/date";  
export \* from "./formatting/email";  
export \* from...

REQ-2481: import { t } from "ttag";  
  
import { color } from "metabase/lib/colors";  
import type { Group } from "metabase-types/api";  
  
const SPECIAL\_GROUP\_NAMES = new Map([  
 // eslint-disable-next-line...

REQ-2482: import { HAS\_LOCAL\_STORAGE } from "metabase/lib/dom";  
  
// If enabled this monkeypatches `t` and `jt` to return blacked out  
// strings/elements to assist in finding untranslated strings.  
//  
//...

REQ-2483: import dayjs from "dayjs";  
import moment from "moment-timezone"; // eslint-disable-line no-restricted-imports -- deprecated usage  
import { addLocale, useLocale } from "ttag";  
  
import { isEmbeddingSdk...

REQ-2484: import moment from "moment-timezone"; // eslint-disable-line no-restricted-imports -- deprecated usage  
  
import { setLocalization } from "./i18n";  
  
function setup(language: string) {  
 ...

REQ-2485: import { PERSONAL\_COLLECTIONS } from "metabase/entities/collections/constants";  
import { PLUGIN\_COLLECTIONS } from "metabase/plugins";  
import type { IconName } from "metabase/ui";  
import {...

REQ-2486: import register from "metabase/visualizations/register";  
  
import { getIcon } from "./icon";  
  
describe("getIcon", () => {  
 beforeAll(() => {  
 // visualizations are weird and we have to register...

REQ-2487: export const safeJsonParse = (value: string | null | undefined) => {  
 if (!value) {  
 return null;  
 }  
  
 try {  
 return JSON.parse(value);  
 } catch (e) {  
 console.error("Unable to parse...

REQ-2488: export const KEYCODE\_SPACE = 0;  
export const KEYCODE\_BACKSPACE = 8;  
export const KEYCODE\_TAB = 9;  
export const KEYCODE\_ENTER = 13;  
export const KEYCODE\_ESCAPE = 27;  
  
export const KEYCODE\_LEFT =...

REQ-2489: export const openImageBlobOnStorybook = ({  
 canvas,  
 blob,  
}: {  
 canvas: HTMLCanvasElement;  
 blob: Blob;  
}) => {  
 const imgElement = document.createElement("img");  
 imgElement.src =...

REQ-2490: import \_ from "underscore";  
  
import type {  
 FontStyle,  
 TextMeasurer,  
} from "metabase/visualizations/shared/types/measure-text";  
  
let canvas: HTMLCanvasElement | null = null;  
  
export const...

REQ-2491: type ItemWithName = {  
 name: string;  
 display\_name?: string;  
};  
  
/\*\* items may or may not have display names, but we should prefer displaying them when they do \*/  
export const getName = (item:...

REQ-2492: there's nothing here!  
This file is the alternative to importing EE plugins.  
Either way we import something to ensure a consisten import order.

REQ-2493: import { c, msgid, ngettext, t } from "ttag";  
import \_ from "underscore";  
  
import type { NotificationListItem } from "metabase/account/notifications/types";  
import { cronToScheduleSettings } from...

REQ-2494: const INTEGER\_REGEX = /^[+-]?\d+$/;  
  
export type NumberValue = number | bigint;  
  
export function parseNumber(value: string): NumberValue | null {  
 const number = parseFloat(value);  
 if...

REQ-2495: import { parseNumber, parseNumberValue } from "./number";  
  
describe("metabase/lib/number", () => {  
 describe("parseNumberValue", () => {  
 it("should return null for non-number values", () => {  
 ...

REQ-2496: export const getObjectEntries = <K extends string, V>(  
 obj: Record<K, V>,  
): [K, V][] => {  
 return Object.entries(obj) as [K, V][];  
};  
  
export const getObjectKeys = <K extends string>(  
 obj:...

REQ-2497: import { noop } from "underscore";  
  
type CancellablePromise<T> = Promise<T> & {  
 cancel: () => void;  
};  
  
// return a promise wrapping the provided one but with a "cancel" method  
export function...

REQ-2498: import { msgid, ngettext, t } from "ttag";  
import \_ from "underscore";  
  
import { getEmailDomain } from "metabase/lib/email";  
import { formatDateTimeWithUnit } from...

REQ-2499: Support React 17 backwards compatibility for the Embedding SDK

REQ-2500: import type { AnyAction, Store, ThunkDispatch } from "@reduxjs/toolkit";  
import type { TypedUseSelectorHook } from "react-redux";  
import {  
 createDispatchHook,  
 createSelectorHook,  
 ...

REQ-2501: export \* from "./utils";  
export \* from "./typed-utils";  
export \* from "./hooks";  
export \* from "./custom-context";

REQ-2502: import type { ThunkDispatch } from "@reduxjs/toolkit";  
import { createAsyncThunk as createAsyncThunkOriginal } from "@reduxjs/toolkit";  
  
import type { GetState, State } from...

REQ-2503: import { compose } from "@reduxjs/toolkit";  
import { getIn } from "icepick";  
import { normalize } from "normalizr";  
import \_ from "underscore";  
  
import { delay } from "metabase/lib/promise";  
import...

REQ-2504: import { delay } from "\_\_support\_\_/utils";  
  
import { fetchData, mergeEntities, updateData } from "./utils";  
  
describe("Metadata", () => {  
 const getDefaultArgs = ({  
 existingData = "data",  
 ...

REQ-2505: import { ResizeObserver as JuggleResizeObserver } from "@juggle/resize-observer";  
  
type ResizeObserverCallback = (  
 entry: ResizeObserverEntry,  
 observer: ResizeObserver,  
) => void;  
  
// PR:...

REQ-2506: export \* from "./schema";

REQ-2507: backend returns model = "card" instead of "question"

REQ-2508: import { entityTypeForModel, entityTypeForObject } from "./schema";  
  
describe("schemas", () => {  
 const MODEL\_ENTITY\_TYPE = [  
 { model: "card", entityType: "questions" },  
 { model: "dataset",...

REQ-2509: import { FIELD\_SEMANTIC\_TYPES\_MAP } from "metabase/lib/core";  
  
export function foreignKeyCountsByOriginTable(fks) {  
 if (fks === null || !Array.isArray(fks)) {  
 return null;  
 }  
  
 return fks  
 ...

REQ-2510: import passwordGenerator from "password-generator";  
  
import MetabaseSettings from "metabase/lib/settings";  
  
// generate a password that satisfies `complexity` requirements, by default the ones that...

REQ-2511: import MetabaseSettings from "metabase/lib/settings";  
  
import { generatePassword } from "./security";  
  
describe("generatePassword", () => {  
 it("defaults to at least 14 characters even if...

REQ-2512: import { msgid, ngettext, t } from "ttag";  
import \_ from "underscore";  
  
import { numberToWord } from "metabase/lib/utils";  
import type {  
 PasswordComplexity,  
 SettingKey,  
 Settings,  
} from...

REQ-2513: import \_ from "underscore";  
  
// Creates a regex that will find an order dependent, case insensitive substring. All whitespace will be rendered as ".\*" in the regex, to create a fuzzy search.  
export...

REQ-2514: export const isSyncInProgress = (entity) => {  
 return entity.initial\_sync\_status === "incomplete";  
};  
  
export const isSyncCompleted = (entity) => {  
 return entity.initial\_sync\_status ===...

REQ-2515: import type { Dayjs } from "dayjs";  
import dayjs from "dayjs";  
import { t } from "ttag";  
  
import type { DatetimeUnit } from "metabase-types/api/query";  
  
const DAYLIGHT\_SAVINGS\_CHANGE\_TOLERANCE:...

REQ-2516: import dayjs from "dayjs";  
  
import { parseTimestamp } from "metabase/lib/time-dayjs";  
  
describe("parseTimestamp", () => {  
 afterEach(() => {  
 dayjs.updateLocale(dayjs.locale(), { weekStart: 0...

REQ-2517: import moment from "moment-timezone"; // eslint-disable-line no-restricted-imports -- deprecated usage  
import { t } from "ttag";  
  
import MetabaseSettings from "metabase/lib/settings";  
import type {...

REQ-2518: import moment from "moment-timezone"; // eslint-disable-line no-restricted-imports -- deprecated usage  
  
import { parseTime } from "./time";  
  
describe("parseTime", () => {  
 it("should return a moment...

REQ-2519: import { t } from "ttag";  
import \_ from "underscore";  
  
import { canonicalCollectionId } from "metabase/collections/utils";  
import type { IconName } from "metabase/ui";  
import type { Collection,...

REQ-2520: import \_ from "underscore";  
  
export const isNotNull = <T>(value: T | null | undefined): value is T => {  
 return value != null;  
};  
  
export const isNotFalsy = <T>(  
 value: T | null | undefined |...

REQ-2521: import { removeNullAndUndefinedValues } from "./types";  
  
describe("removeNullAndUndefinedValues", () => {  
 it("removes nil values from an object", () => {  
 const obj = {  
 a: 1,  
 b:...

REQ-2522: import type { FileUpload } from "metabase-types/store/upload";  
  
export const isUploadInProgress = (upload: FileUpload) =>  
 upload.status === "in-progress";  
  
export const isUploadCompleted = (upload:...

REQ-2523: import type { CardId, WritebackActionId } from "metabase-types/api";  
  
import { modelDetail } from "./models";  
  
type ParentModelProps = {  
 id: CardId;  
 name?: string;  
};  
  
export function...

REQ-2524: import type { DatabaseId, UserId } from "metabase-types/api";  
  
export function newUser() {  
 return `/admin/people/new`;  
}  
  
export function editUser(userId: UserId) {  
 return...

REQ-2525: export const login = (redirectUrl?: string) => {  
 return redirectUrl  
 ? `/auth/login?redirect=${encodeURIComponent(redirectUrl)}`  
 : "/auth/login";  
};  
  
export const password = (redirectUrl?:...

REQ-2526: import slugg from "slugg";  
  
import type { Bookmark } from "metabase-types/api";  
  
import { appendSlug } from "./utils";  
  
function getBookmarkBasePath(bookmark: Pick<Bookmark, "type" | "card\_type">) {  
...

REQ-2527: import slugg from "slugg";  
  
import type DatabaseV1 from "metabase-lib/v1/metadata/Database";  
import type Table from "metabase-lib/v1/metadata/Table";  
import { SAVED\_QUESTIONS\_VIRTUAL\_DB\_ID } from...

REQ-2528: import slugg from "slugg";  
  
import {  
 isRootPersonalCollection,  
 isRootTrashCollection,  
} from "metabase/collections/utils";  
import type {  
 Collection as BaseCollection,  
 CollectionId,  
} from...

REQ-2529: import slugg from "slugg";  
  
import { stringifyHashOptions } from "metabase/lib/browser";  
import MetabaseSettings from "metabase/lib/settings";  
import type {  
 DashCardId,  
 DashboardId,  
 ...

REQ-2530: export \* from "./actions";  
export \* from "./admin";  
export \* from "./auth";  
export \* from "./bookmarks";  
export \* from "./browse";  
export \* from "./collections";  
export \* from "./dashboards";  
export...

REQ-2531: import slugg from "slugg";  
  
import type { IndexedEntity } from "metabase-types/api";  
  
export const indexedEntity = (entity: IndexedEntity) =>  
 ...

REQ-2532: import type {  
 ExportFormat,  
 TableExportFormat,  
} from "metabase/common/types/export";  
  
export const exportFormats: TableExportFormat[] = ["csv", "xlsx", "json"];  
export const exportFormatPng:...

REQ-2533: import { collection } from "./collections";  
import { dashboard } from "./dashboards";  
import { metric, model } from "./models";  
import { question, tableRowsQuery } from "./questions";  
  
type...

REQ-2534: import { modelToUrl } from "./modelToUrl";  
  
describe("urls > modelToUrl", () => {  
 it("should return null for unknown model", () => {  
 expect(  
 // @ts-expect-error - testing the error case  
 ...

REQ-2535: import slugg from "slugg";  
  
import type { Card } from "metabase-types/api";  
  
import type { QuestionUrlBuilderParams } from "./questions";  
import { question } from "./questions";  
import { appendSlug }...

REQ-2536: export function pulse(pulseId: number) {  
 return `/pulse/${pulseId}`;  
}  
  
export function pulseEdit(pulseId: number) {  
 return `/pulse/${pulseId}`;  
}

REQ-2537: import slugg from "slugg";  
  
import { serializeCardForUrl } from "metabase/lib/card";  
import MetabaseSettings from "metabase/lib/settings";  
import type { QuestionCreatorOpts } from...

REQ-2538: import type { Collection, Timeline, TimelineEvent } from "metabase-types/api";  
  
import { collection as getCollectionUrl } from "./collections";  
  
export function timelinesInCollection(collection?:...

REQ-2539: import api from "metabase/lib/api";  
  
export function appendSlug(path: string | number, slug?: string) {  
 return slug ? `${path}-${slug}` : String(path);  
}  
  
export function extractEntityId(slug = "")...

REQ-2540: import api from "metabase/lib/api";  
  
import { getSubpathSafeUrl, openInNewTab } from "./utils";  
  
const fakeBasename = "foobar";  
const originalBasename = api.basename;  
  
const mockWindowOpen =...

REQ-2541: export function getFullName(user: NamedUser): string | null {  
 const firstName = user.first\_name?.trim() || "";  
 const lastName = user.last\_name?.trim() || "";  
 return [firstName, lastName].join("...

REQ-2542: import { t } from "ttag";  
import \_ from "underscore";  
  
import { PLUGIN\_IS\_EE\_BUILD } from "metabase/plugins";  
  
export function isEmpty(str: string | null) {  
 if (str != null) {  
 str =...

REQ-2543: export function uuid() {  
 return (  
 s4() +  
 s4() +  
 "-" +  
 s4() +  
 "-" +  
 s4() +  
 "-" +  
 s4() +  
 "-" +  
 s4() +  
 s4() +  
 s4()  
 );  
}  
  
function s4() {  
 return...

REQ-2544: import { t } from "ttag";  
  
import { isEmail } from "metabase/lib/email";  
import Settings from "metabase/lib/settings";  
  
// we need this to allow 0 as a valid form value  
export const isEmpty = (value)...

REQ-2545: import { type RefObject, useContext, useEffect, useMemo } from "react";  
  
import { PLUGIN\_METABOT } from "metabase/plugins";  
import type { MetabotChatContext } from "metabase-types/api";  
import type {...

REQ-2546: export \* from "./context";

REQ-2547: import { createSelector } from "@reduxjs/toolkit";  
import \_ from "underscore";  
  
import {  
 getRawSeries,  
 getTransformedSeries,  
 getVisualizationSettings,  
} from...

REQ-2548: {  
 "rules": {  
 // Note: adding this rule to a eslint config file in a subfolder will remove  
 // \*not\* carry over the restricted imports from parent folders, you will  
 // need to copy them...

REQ-2549: Metadata

REQ-2550: export \* from "./CoercionStrategyPicker";

REQ-2551: import { c, t } from "ttag";  
  
const GET\_LEFT\_TERM\_CONVERSIONS = (): Record<string, string> => ({  
 ISO8601: t`ISO 8601`,  
 UNIXSeconds: t`UNIX seconds`,  
 UNIXMilliSeconds: t`UNIX milliseconds`,  
 ...

REQ-2552: import { humanizeCoercionStrategy } from "./utils";  
  
describe("humanizeCoercionStrategy", () => {  
 it("does not convert `Don't cast`", () => {  
 const original = "Don't cast";  
 const humanized...

REQ-2553: export \* from "./CurrencyPicker";

REQ-2554: export \* from "./DiscardFieldValuesButton";

REQ-2555: export \* from "./DiscardTableFieldValuesButton";

REQ-2556: export \* from "./FieldOrderPicker";

REQ-2557: export \* from "./FieldValuesTypePicker";

REQ-2558: export \* from "./FieldVisibilityPicker";

REQ-2559: export \* from "./FkTargetPicker";

REQ-2560: export \* from "./NameDescriptionInput";

REQ-2561: export \* from "./RescanFieldButton";

REQ-2562: export \* from "./RescanTableFieldsButton";

REQ-2563: export { SemanticTypeAndTargetPicker } from "./SemanticTypeAndTargetPicker";

REQ-2564: export \* from "./SemanticTypePicker";

REQ-2565: import { FIELD\_SEMANTIC\_TYPES } from "metabase/lib/core";  
import { LEVEL\_ONE\_TYPES, TYPE } from "metabase-lib/v1/types/constants";  
import { isTypeFK, isTypePK, isa } from...

REQ-2566: export \* from "./SortableFieldItem";

REQ-2567: export \* from "./SortableFieldList";

REQ-2568: export \* from "./SyncTableSchemaButton";

REQ-2569: export \* from "./TableBreadcrumbs";

REQ-2570: export \* from "./UnfoldJsonPicker";

REQ-2571: export \* from "./CoercionStrategyPicker";  
export \* from "./CurrencyPicker";  
export \* from "./DiscardFieldValuesButton";  
export \* from "./DiscardTableFieldValuesButton";  
export \* from...

REQ-2572: export \* from "./useMetadataToasts";

REQ-2573: import { useCallback } from "react";  
import { t } from "ttag";  
  
import { useToast } from "metabase/common/hooks";  
  
export const useMetadataToasts = () => {  
 const [sendToast] = useToast();  
  
 const...

REQ-2574: export \* from "./CustomMappingModal";  
export \* from "./types";

REQ-2575: export type Mapping = Map<number, string>;  
  
export interface ChangeOptions {  
 isAutomatic?: boolean;  
}

REQ-2576: import type { Mapping } from "./types";  
  
export function areMappingsEqual(a: Mapping, b: Mapping): boolean {  
 return (  
 a.size === b.size && [...a].every(([key, value]) => b.get(key) === value)  
 ...

REQ-2577: export \* from "./DisplayValuesPicker";  
export \* from "./types";

REQ-2578: export type RemappingValue = "original" | "foreign" | "custom";

REQ-2579: export \* from "./RemappingPicker";

REQ-2580: import { t } from "ttag";  
import \_ from "underscore";  
  
import { getColumnIcon } from "metabase/common/utils/columns";  
import { getRawTableFieldId } from "metabase/metadata/utils/field";  
import \* as...

REQ-2581: export \* from "./BehaviorSection";

REQ-2582: export \* from "./DataSection";

REQ-2583: import { useElementSize } from "@mantine/hooks";  
import { useState } from "react";  
  
const BUTTONS\_GAP = 16;  
  
export const useResponsiveButtons = () => {  
 const { ref: buttonsContainerRef, width:...

REQ-2584: export \* from "./FieldSection";

REQ-2585: export \* from "./Error";

REQ-2586: export \* from "./PreviewSection";  
export \* from "./types";

REQ-2587: export type PreviewType = "table" | "detail" | "filtering";

REQ-2588: import { t } from "ttag";  
  
import type { Dataset } from "metabase-types/api";  
import { isObject } from "metabase-types/guards";  
  
export function getPreviewTypeData() {  
 return [  
 { label:...

REQ-2589: export \* from "./SegmentsLink";

REQ-2590: export \* from "./TablePicker";  
export \* from "./wrappers";

REQ-2591: import type {  
 DatabaseId,  
 SchemaName,  
 Table,  
 TableId,  
} from "metabase-types/api";  
  
export type NodeKey = string;  
  
export type TreePath = {  
 databaseId?: DatabaseId;  
 schemaName?:...

REQ-2592: import { useCallback, useEffect, useMemo, useState } from "react";  
import { useDeepCompareEffect, useLatest } from "react-use";  
import \_ from "underscore";  
  
import {  
 skipToken,  
 ...

REQ-2593: export \* from "./FieldItem";

REQ-2594: export \* from "./FieldList";

REQ-2595: import { useElementSize } from "@mantine/hooks";  
import { useState } from "react";  
  
const BUTTONS\_GAP = 16;  
const LOADER\_WIDTH = 16;  
const FIELD\_ORDER\_PICKER\_WIDTH = 136;  
  
export const...

REQ-2596: export \* from "./TableSection";

REQ-2597: export \* from "./TitledSection";

REQ-2598: export \* from "./FieldSection";  
export \* from "./FieldValuesModal";  
export \* from "./NoDatabasesEmptyState";  
export \* from "./PreviewSection";  
export \* from "./ResponsiveButton";  
export \* from...

REQ-2599: import type { Column, ColumnSizeConfig } from "./types";  
  
const PREVIEW\_COLUMN\_PADDING = 2 \* 32;  
  
export const COLUMN\_CONFIG: Record<Column, ColumnSizeConfig> = {  
 nav: {  
 flex: "6 1 0",  
 min:...

REQ-2600: export \* from "./DataModel";

REQ-2601: import type {  
 DatabaseId,  
 FieldId,  
 SchemaId,  
 SchemaName,  
 TableId,  
} from "metabase-types/api";  
  
export type RouteParams = {  
 databaseId?: string;  
 fieldId?: string;  
 schemaId?:...

REQ-2602: import { skipToken } from "metabase/api";  
import \* as Urls from "metabase/lib/urls";  
import { PLUGIN\_FEATURE\_LEVEL\_PERMISSIONS } from "metabase/plugins";  
// eslint-disable-next-line...

REQ-2603: import type { ParsedRouteParams, RouteParams } from "./types";  
import { getUrl, parseRouteParams } from "./utils";  
  
describe("parseRouteParams", () => {  
 it("should parse all route parameters...

REQ-2604: import type { Database, DatabaseFeature } from "metabase-types/api";  
  
export function hasDatabaseFeature(  
 database: Database,  
 feature: DatabaseFeature | string | null | undefined,  
): boolean {  
 ...

REQ-2605: import { createMockDatabase } from "metabase-types/api/mocks";  
  
import { hasDatabaseFeature } from "./database";  
  
describe("hasDatabaseFeature", () => {  
 it("returns true when feature is null", ()...

REQ-2606: import { is\_coerceable } from "cljs/metabase.types.core";  
import { NULL\_DISPLAY\_VALUE } from "metabase/lib/constants";  
import { getGlobalSettingsForColumn } from...

REQ-2607: import { NULL\_DISPLAY\_VALUE } from "metabase/lib/constants";  
import MetabaseSettings from "metabase/lib/settings";  
import {  
 createMockDatabase,  
 createMockField,  
 createMockTable,  
} from...

REQ-2608: import { humanize, titleize } from "metabase/lib/formatting";  
import type { SchemaName } from "metabase-types/api";  
  
export function getSchemaDisplayName(schema: SchemaName): string {  
 return...

REQ-2609: import { getSchemaDisplayName } from "./schema";  
  
describe("getSchemaDisplayName", () => {  
 it("should handle empty string", () => {  
 expect(getSchemaDisplayName("")).toBe("");  
 });  
  
 ...

REQ-2610: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2611: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2612: export \* from "./FormModelPicker";

REQ-2613: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2614: export \* from "./AdminNavbar";

REQ-2615: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2616: import { isRootCollection } from "metabase/collections/utils";  
import type { Collection, CollectionId } from "metabase-types/api";  
  
type GetCollectionListProps = {  
 collection: Collection;  
 ...

REQ-2617: import type { CollectionEssentials, CollectionId } from "metabase-types/api";  
import {  
 createMockCollection,  
 createMockCollectionEssential,  
} from "metabase-types/api/mocks";  
  
import {...

REQ-2618: export \* from "./DevModeBanner";

REQ-2619: export { LicenseTokenMissingBanner } from "./LicenseTokenMissingBanner";  
export { useLicenseTokenMissingBanner } from "./useLicenseTokenMissingBanner";

REQ-2620: import dayjs from "dayjs";  
import utc from "dayjs/plugin/utc";  
  
import { useUpdateSettingMutation } from "metabase/api";  
import { useSetting } from "metabase/common/hooks";  
import { isEEBuild } from...

REQ-2621: import { act } from "@testing-library/react";  
  
import { setupEnterprisePlugins } from "\_\_support\_\_/enterprise";  
import {  
 findRequests,  
 setupPropertiesEndpoints,  
 setupSettingsEndpoints,  
 ...

REQ-2622: import { trackSimpleEvent } from "metabase/lib/analytics";  
  
export const trackAppNewButtonClicked = () =>  
 trackSimpleEvent({  
 event: "new\_button\_clicked",  
 triggered\_from: "app-bar",  
 });

REQ-2623: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2624: export { default as ProfileLink } from "./ProfileLink";

REQ-2625: import { useEffect, useState } from "react";  
  
import { useSetting } from "metabase/common/hooks";  
import { useSelector } from "metabase/lib/redux";  
import { getIsPaidPlan } from...

REQ-2626: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2627: export \* from "./ReadOnlyBanner";

REQ-2628: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2629: export \* from "./TrialBanner";

REQ-2630: export { WhatsNewNotification } from "./WhatsNewNotification";

REQ-2631: import { isNotFalsy } from "metabase/lib/types";  
import { compareVersions } from "metabase/lib/utils";  
import type { VersionInfoRecord } from "metabase-types/api";  
import type { VersionInfo } from...

REQ-2632: import { createMockVersionInfoRecord as mockVersion } from "metabase-types/api/mocks";  
import type {  
 VersionInfo,  
 VersionInfoRecord,  
} from "metabase-types/api/settings";  
  
import {...

REQ-2633: export { RecentsList } from "./RecentsList";

REQ-2634: import { isSyncCompleted } from "metabase/lib/syncing";  
import \* as Urls from "metabase/lib/urls";  
import type { RecentItem } from "metabase-types/api";  
  
export const isItemActive = (item:...

REQ-2635: export { default as SearchBar } from "./SearchBar";

REQ-2636: export \* from "./SearchButton";

REQ-2637: export { SearchResults, SearchLoadingSpinner } from "./SearchResults";  
export type { SearchResultsFooter } from "./SearchResults";  
export \* from "./SearchResults.styled";

REQ-2638: export const MIN\_RESULTS\_FOR\_FOOTER\_TEXT = 4;

REQ-2639: export { SearchResultsDropdown } from "./SearchResultsDropdown";

REQ-2640: export const APP\_BAR\_HEIGHT = "52px";  
export const APP\_SUBHEADER\_HEIGHT = "48px";  
export const APP\_BAR\_EXTENDED\_HEIGHT = "98px";  
export const ADMIN\_NAVBAR\_HEIGHT = "65px";  
export const...

REQ-2641: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2642: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2643: import { trackSimpleEvent } from "metabase/lib/analytics";  
import type { Engine } from "metabase-types/api";  
export const trackAddDataEvent = (  
 event: "csv\_tab\_clicked" | "sheets\_tab\_clicked" |...

REQ-2644: export { AddDataModal } from "./AddDataModal";

REQ-2645: const validTabArray = ["db", "csv", "gsheets"] as const;  
const validTabs = new Set<string>(validTabArray);  
  
type AddDataTab = (typeof validTabArray)[number];  
  
export const isValidTab = (v: string |...

REQ-2646: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2647: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2648: export { DraggableSidebarLink } from "./DraggableSidebarLink";  
export { default as SidebarCollectionLink } from "./SidebarCollectionLink";  
export { default as SidebarLink } from "./SidebarLink";

REQ-2649: import { trackSimpleEvent } from "metabase/lib/analytics";  
  
export const trackOnboardingChecklistOpened = () => {  
 trackSimpleEvent({  
 event: "onboarding\_checklist\_opened",  
 });  
};  
  
export const...

REQ-2650: import { coerceCollectionId } from "metabase/collections/utils";  
import \* as Urls from "metabase/lib/urls";  
import type Question from "metabase-lib/v1/Question";  
import type { Collection, Dashboard }...

REQ-2651: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2652: import type { Location } from "history";  
  
export interface MainNavbarOwnProps {  
 isOpen: boolean;  
 location: Location;  
 params: {  
 slug?: string;  
 pageId?: string;  
 };  
}  
  
export interface...

REQ-2653: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2654: export { CaveatMessage } from "./CaveatMessage";

REQ-2655: import { screen } from "\_\_support\_\_/ui";  
  
import { setup } from "./setup";  
  
describe("CaveatMessage (OSS)", () => {  
 it("should show a help link when `show-metabase-links: true`", () => {  
 ...

REQ-2656: import { screen } from "\_\_support\_\_/ui";  
  
import type { SetupOpts } from "./setup";  
import { setup as baseSetup } from "./setup";  
  
function setup(opts: SetupOpts) {  
 baseSetup({...

REQ-2657: import { screen } from "\_\_support\_\_/ui";  
  
import type { SetupOpts } from "./setup";  
import { setup as baseSetup } from "./setup";  
  
function setup(opts: SetupOpts) {  
 baseSetup({  
 ...

REQ-2658: import { t } from "ttag";  
  
import type { ChannelType } from "metabase-types/api";  
  
export const CHANNEL\_NOUN\_PLURAL: Record<ChannelType, string> = {  
 get email() {  
 return t`Emails`;  
 },  
 get...

REQ-2659: import {  
 createMockCard,  
 createMockDashboard,  
 createMockDashboardCard,  
} from "metabase-types/api/mocks";  
  
import { getSupportedCardsForSubscriptions } from...

REQ-2660: import userEvent from "@testing-library/user-event";  
import fetchMock from "fetch-mock";  
  
import { screen } from "\_\_support\_\_/ui";  
  
import { dashcard, hasBasicFilterOptions, setup, user } from...

REQ-2661: import userEvent from "@testing-library/user-event";  
  
import { screen } from "\_\_support\_\_/ui";  
  
import { hasBasicFilterOptions, setup } from "./setup";  
  
describe("DashboardSubscriptionsSidebar...

REQ-2662: import userEvent from "@testing-library/user-event";  
  
import { screen, within } from "\_\_support\_\_/ui";  
import {  
 createMockCard,  
 createMockDashboardCard,  
 createMockParameter,  
} from...

REQ-2663: import { mockSettings } from "\_\_support\_\_/settings";  
import { fireEvent, renderWithProviders, screen } from "\_\_support\_\_/ui";  
import { createMockState } from "metabase-types/store/mocks";  
  
import...

REQ-2664: export \* from "./DashboardSubscriptionMenuItem";  
export \* from "./QuestionAlertsMenuItem";

REQ-2665: export { CreateOrEditQuestionAlertModal } from "./CreateOrEditQuestionAlertModal";

REQ-2666: import type { NotificationCardSendCondition } from "metabase-types/api";  
  
// TODO: combine this with api types  
export type NotificationTriggerOption = {  
 value: NotificationCardSendCondition;  
 ...

REQ-2667: export { QuestionAlertListModal } from "./QuestionAlertListModal";

REQ-2668: export { CreateOrEditQuestionAlertModal } from "./CreateOrEditQuestionAlertModal";  
export { QuestionAlertListModal } from "./QuestionAlertListModal";

REQ-2669: import { createAction } from "redux-actions";  
import { t } from "ttag";  
  
import { getActionErrorMessage } from "metabase/actions/utils";  
import Pulses from "metabase/entities/pulses";  
import {  
 ...

REQ-2670: import { handleActions } from "redux-actions";  
  
import {  
 CANCEL\_EDITING\_PULSE,  
 FETCH\_PULSE\_CARD\_PREVIEW,  
 FETCH\_PULSE\_FORM\_INPUT,  
 FETCH\_PULSE\_LIST\_BY\_DASHBOARD\_ID,  
 SAVE\_EDITING\_PULSE,  
 ...

REQ-2671: export const getEditingPulse = (state) => state.pulse.editingPulse;  
  
export const getPulseFormInput = (state) => state.pulse?.formInput;

REQ-2672: import type {  
 CardId,  
 ChannelApiResponse,  
 CreateAlertNotificationRequest,  
 NotificationChannel,  
 NotificationHandler,  
 ScheduleSettings,  
 UserId,  
} from "metabase-types/api";  
  
import type {...

REQ-2673: export \* from "./PaletteShortcutsModal";

REQ-2674: import { t } from "ttag";  
  
import type { Settings } from "metabase-types/api";  
  
export const GROUP\_LABELS = {  
 get global() {  
 return t`General`;  
 },  
 get dashboard() {  
 return...

REQ-2675: import { t } from "ttag";  
  
import { ELLIPSIS } from "../constants";  
  
export const adminShortcuts = {  
 "admin-change-tab": {  
 get name() {  
 return t`Change admin tab`;  
 },  
 shortcut:...

REQ-2676: import { t } from "ttag";  
  
export const collectionShortcuts = {  
 "collection-send-items-to-trash": {  
 get name() {  
 return t`Move collection items to trash`;  
 },  
 shortcut:...

REQ-2677: import { t } from "ttag";  
  
import { ELLIPSIS } from "../constants";  
  
export const dashboardShortcuts = {  
 "dashboard-bookmark": {  
 get name() {  
 return t`Bookmark dashboard`;  
 },  
 ...

REQ-2678: import { t } from "ttag";  
  
export const globalShortcuts = {  
 "create-new-question": {  
 get name() {  
 return t`Create a question`;  
 },  
 shortcut: ["c q"],  
 shortcutGroup: "global" as...

REQ-2679: import { adminShortcuts } from "./admin";  
import { collectionShortcuts } from "./collection";  
import { dashboardShortcuts } from "./dashboard";  
import { globalShortcuts } from "./global";  
import {...

REQ-2680: import { t } from "ttag";  
  
export const questionShortcuts = {  
 "query-builder-toggle-notebook-editor": {  
 get name() {  
 return t`Switch to editor`;  
 },  
 shortcut: ["e"],  
 ...

REQ-2681: import { type KeyboardShortcutId, shortcuts } from ".";  
  
const getShortcutsWithPrefix = (prefix: string) => {  
 const allIds = Object.keys(shortcuts) as KeyboardShortcutId[];  
 return allIds  
 ...

REQ-2682: import type { LocationDescriptor } from "history";  
import type { Action, ActionImpl } from "kbar";  
import type React from "react";  
  
import type { IconName } from "metabase/ui";  
import type {...

REQ-2683: import type { LocationDescriptor } from "history";  
import type { MouseEvent } from "react";  
import { t } from "ttag";  
import \_ from "underscore";  
  
import { color } from "metabase/lib/colors";  
import...

REQ-2684: import type { LocationDescriptor } from "history";  
  
import type { PaletteActionImpl } from "./types";  
import {  
 locationDescriptorToURL,  
 navigateActionIndex,  
 processResults,  
 processSection,  
}...

REQ-2685: import { CardApi, DashboardApi, ParameterApi } from "metabase/services";  
import { getNonVirtualFields } from "metabase-lib/v1/parameters/utils/parameter-fields";  
import { normalizeParameter } from...

REQ-2686: export { FilterApplyToast } from "./FilterApplyToast";

REQ-2687: import { msgid, ngettext } from "ttag";  
import \_ from "underscore";  
  
function isParameterValueEmpty(value: unknown): boolean {  
 return value == null || (Array.isArray(value) && value.length ===...

REQ-2688: import { getFilterChangeDescription } from "./utils";  
  
describe("getFilterChangeDescription", () => {  
 describe("single filter changes", () => {  
 it("should describe a single filter added", () =>...

REQ-2689: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2690: export { ParameterLinkedFilters } from "./ParameterLinkedFilters";

REQ-2691: import type { UiParameter } from "metabase-lib/v1/parameters/types";  
import type { FieldId } from "metabase-types/api";  
  
export type ParameterInfo = {  
 parameter: UiParameter;  
 filteredIds:...

REQ-2692: import type { UiParameter } from "metabase-lib/v1/parameters/types";  
import { getFields } from "metabase-lib/v1/parameters/utils/parameter-fields";  
import type {  
 FieldId,  
 ...

REQ-2693: import { checkNotNull } from "metabase/lib/types";  
import { SAMPLE\_METADATA } from "metabase-lib/test-helpers";  
import { createMockUiParameter } from "metabase-lib/v1/parameters/mock";  
import type {...

REQ-2694: export \* from "./MoveParameterMenu";

REQ-2695: export \* from "./TemporalUnitSettings";

REQ-2696: export { ParameterSettings } from "./ParameterSettings";

REQ-2697: export { ParameterSidebar } from "./ParameterSidebar";

REQ-2698: export \* from "./ParameterWidget";

REQ-2699: export \* from "./ParametersList";  
export \* from "./types";

REQ-2700: import type {  
 DashboardFullscreenControls,  
 DashboardNightModeControls,  
 EmbedHideParametersControls,  
} from "metabase/dashboard/types";  
import type Question from...

REQ-2701: export { RequiredParamToggle } from "./RequiredParamToggle";

REQ-2702: import { t } from "ttag";  
  
import { areParameterValuesIdentical } from "metabase-lib/v1/parameters/utils/parameter-values";  
  
const getUpdateLabel = () => t`Update filter`;  
const getAddLabel = () =>...

REQ-2703: import { getUpdateButtonProps } from "./getUpdateButtonProps";  
  
describe("getUpdateButtonProps", () => {  
 describe("non-required parameters", () => {  
 it("without both value and unsaved, shows...

REQ-2704: export { UpdateFilterButton } from "./UpdateFilterButton";

REQ-2705: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2706: import { parse } from "csv-parse/browser/esm/sync";  
import { stringify } from "csv-stringify/browser/esm/sync";  
  
import type { ParameterValue } from "metabase-types/api";  
  
export const getValuesText...

REQ-2707: import { getStaticValues, getValuesText } from "./utils";  
  
describe("getValuesText", () => {  
 it("should stringify just values correctly", () => {  
 expect(getValuesText(["Foo", "Bar",...

REQ-2708: export { ValuesSourceSettings } from "./ValuesSourceSettings";

REQ-2709: export \* from "./WidgetStatus";

REQ-2710: \*

REQ-2711: export { NumberInputWidget } from "./NumberInputWidget";

REQ-2712: export { ListField } from "./ListField";

REQ-2713: import \_ from "underscore";  
  
import type { ContentTranslationFunction } from "metabase/i18n/types";  
import type { DictionaryResponse } from "metabase-types/api";  
  
export const translateToGerman:...

REQ-2714: import type { JSX } from "react";  
  
import type { FieldValue, RowValue } from "metabase-types/api";  
export type Option = FieldValue;  
  
export const getOptionDisplayName = (option: Option | RowValue[])...

REQ-2715: import type { ContentTranslationFunction } from "metabase/i18n/types";  
import type { FieldValue, RowValue } from "metabase-types/api";  
  
import type { Option } from...

REQ-2716: import type { ContentTranslationFunction } from "metabase/i18n/types";  
import type { FieldValue } from "metabase-types/api";  
  
import { translateToGerman, translateToJapanese } from...

REQ-2717: import { waitForElementToBeRemoved } from "@testing-library/react";  
import userEvent from "@testing-library/user-event";  
  
import { render, screen } from "\_\_support\_\_/ui";  
  
import { Value as...

REQ-2718: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2719: import type { FieldValue, RowValue } from "metabase-types/api";  
  
export type Option = FieldValue;  
  
export interface SingleSelectListFieldProps {  
 onChange: (value: RowValue[]) => void;  
 value:...

REQ-2720: export function optionItemEqualsFilter(  
 optionItem: any,  
 filter: string,  
): boolean {  
 return String(optionItem) === filter;  
}

REQ-2721: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2722: import { createMockEntitiesState } from "\_\_support\_\_/store";  
import { getMetadata } from "metabase/selectors/metadata";  
import { createMockField } from "metabase-types/api/mocks";  
import {  
 ...

REQ-2723: export type ValuesMode = "search" | "list" | "none";  
export type LoadingStateType = "LOADING" | "LOADED" | "INIT";

REQ-2724: import { t } from "ttag";  
import \_ from "underscore";  
  
import { isTransientId } from "metabase/dashboard/utils";  
import { stripId } from "metabase/lib/formatting";  
import type { ComboboxItem } from...

REQ-2725: import { ORDERS, PEOPLE, PRODUCTS } from "metabase-types/api/mocks/presets";  
  
import {  
 LISTABLE\_FIELD\_WITH\_MANY\_VALUES\_ID,  
 STRING\_PK\_FIELD\_ID,  
 metadata,  
} from "./testMocks";  
import {...

REQ-2726: import { render, screen } from "\_\_support\_\_/ui";  
  
import { ParameterFieldWidgetValue } from "./ParameterFieldWidgetValue";  
  
const value = "A value";  
  
describe("when fields is empty array", () => {  
 ...

REQ-2727: export function normalizeValue(value: unknown) {  
 if (Array.isArray(value)) {  
 return value;  
 }  
  
 return value || value === 0 ? [value] : [];  
}

REQ-2728: import { normalizeValue } from "./normalizeValue";  
  
describe("normalizeValue", () => {  
 it("returns empty array if value is null", () => {  
 const value = null;  
 const expected = [];  
  
 const...

REQ-2729: export { StringInputWidget } from "./StringInputWidget";

REQ-2730: export \* from "./TemporalUnitWidget";

REQ-2731: export { TextWidget } from "./TextWidget";

REQ-2732: import type { ComboboxProps } from "metabase/ui";  
  
export const MIN\_WIDTH = 300;  
export const WIDTH = 380;  
export const COMBOBOX\_PROPS: Partial<ComboboxProps> = {  
 width: 314,  
 position:...

REQ-2733: import { handleActions } from "redux-actions";  
  
import {  
 INITIALIZE,  
 RESET,  
 UPDATE\_DASHBOARD\_AND\_CARDS,  
} from "metabase/dashboard/actions";  
import {  
 API\_UPDATE\_QUESTION,  
 INITIALIZE\_QB,  
 ...

REQ-2734: import type { State } from "metabase-types/store";  
  
export const getParameterValuesCache = (state: State) => {  
 return state.parameters.parameterValuesCache;  
};

REQ-2735: import { t } from "ttag";  
  
import {  
 BOOLEAN\_OPTION,  
 ID\_OPTION,  
} from "metabase-lib/v1/parameters/constants";  
import type { ParameterSectionId } from...

REQ-2736: import \_ from "underscore";  
  
import { getDashboardParameterSections } from "./dashboard-options";  
  
describe("parameters/utils/dashboard-options", () => {  
 describe("getDashboardParameterSections",...

REQ-2737: import \_ from "underscore";  
  
import { isQuestionCard, isQuestionDashCard } from "metabase/dashboard/utils";  
import { slugify } from "metabase/lib/formatting";  
import { isNotNull } from...

REQ-2738: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import {  
 createParameter,  
 getFilteringParameterValuesMap,  
 getUnsavedDashboardUiParameters,  
 hasMapping,  
 hasMatchingParameters,  
 ...

REQ-2739: import { getDateFilterDisplayName } from "metabase/querying/filters/utils/dates";  
import { deserializeDateParameterValue } from "metabase/querying/parameters/utils/parsing";  
import type { Parameter }...

REQ-2740: import { msgid, ngettext } from "ttag";  
  
import { formatValue } from "metabase/lib/formatting";  
import \* as Lib from "metabase-lib";  
import Field from "metabase-lib/v1/metadata/Field";  
import type {...

REQ-2741: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import { checkNotNull } from "metabase/lib/types";  
import { createMockUiParameter } from "metabase-lib/v1/parameters/mock";  
import {  
 ...

REQ-2742: import {  
 FIELD\_FILTER\_PARAMETER\_TYPES,  
 TYPE\_SUPPORTS\_LINKED\_FILTERS,  
} from "metabase-lib/v1/parameters/constants";  
import { getParameterType } from...

REQ-2743: import { t } from "ttag";  
import \_ from "underscore";  
  
import { tag\_names } from "cljs/metabase.parameters.shared";  
import { getColumnIcon } from "metabase/common/utils/columns";  
import {...

REQ-2744: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import \* as Lib from "metabase-lib";  
import {  
 SAMPLE\_METADATA,  
 createQueryWithClauses,  
} from "metabase-lib/test-helpers";  
import...

REQ-2745: export function generateParameterId() {  
 const num = Math.floor(Math.random() \* Math.pow(2, 32));  
 return num.toString(16);  
}

REQ-2746: import { generateParameterId } from "./parameter-id";  
  
describe("parameters/utils/parameter-id", () => {  
 it("should generate a random parameter id", () => {  
 ...

REQ-2747: import { SINGLE\_OR\_MULTI\_SELECTABLE\_TYPES } from "metabase-lib/v1/parameters/constants";  
import type { ParameterWithTemplateTagTarget } from "metabase-lib/v1/parameters/types";  
import {  
 ...

REQ-2748: import { createMockUiParameter } from "metabase-lib/v1/parameters/mock";  
import { createMockParameter } from "metabase-types/api/mocks";  
  
import { isSingleOrMultiSelectable } from...

REQ-2749: import \_ from "underscore";  
  
import type { IconName } from "metabase/ui";  
import { isEqualsOperator } from "metabase-lib/v1/operators/utils";  
import type { UiParameter } from...

REQ-2750: import { createMockUiParameter } from "metabase-lib/v1/parameters/mock";  
  
import {  
 buildHiddenParametersSlugSet,  
 getParameterIconName,  
 getParameterWidgetTitle,  
 getVisibleParameters,  
} from...

REQ-2751: import MetabaseSettings from "metabase/lib/settings";  
import {  
 PLUGIN\_AUTH\_PROVIDERS,  
 PLUGIN\_IS\_PASSWORD\_USER,  
} from "metabase/plugins";  
  
PLUGIN\_AUTH\_PROVIDERS.providers.push((providers) => {  
 ...

REQ-2752: import { PLUGIN\_IS\_PASSWORD\_USER } from "metabase/plugins";  
  
PLUGIN\_IS\_PASSWORD\_USER.push((user) => user.sso\_source !== "jwt");

REQ-2753: import { PLUGIN\_IS\_PASSWORD\_USER } from "metabase/plugins";  
  
PLUGIN\_IS\_PASSWORD\_USER.push((user) => user.sso\_source !== "ldap");

REQ-2754: import { PLUGIN\_AUTH\_PROVIDERS } from "metabase/plugins";  
  
PLUGIN\_AUTH\_PROVIDERS.providers.push((providers) => {  
 const passwordProvider = {  
 name: "password",  
 // circular dependencies  
 ...

REQ-2755: import { PLUGIN\_IS\_PASSWORD\_USER } from "metabase/plugins";  
  
PLUGIN\_IS\_PASSWORD\_USER.push((user) => user.sso\_source !== "saml");

REQ-2756: import "metabase/plugins/builtin/auth/password";  
import "metabase/plugins/builtin/auth/google";  
import "metabase/plugins/builtin/auth/ldap";  
import "metabase/plugins/builtin/auth/jwt";  
import...

REQ-2757: export \* from "./PluginPlaceholder";

REQ-2758: import type { Middleware } from "@reduxjs/toolkit";  
import React, {  
 type ComponentType,  
 type Dispatch,  
 type HTMLAttributes,  
 type ReactNode,  
 type SetStateAction,  
 useCallback,  
 useMemo,  
}...

REQ-2759: import type { ComponentType } from "react";  
  
import type { ConfirmationState } from "metabase/common/hooks/use-confirmation";  
import type { Member, Membership, User } from...

REQ-2760: export { LogoBadge } from "./LogoBadge";

REQ-2761: export { EmbedFrame } from "./EmbedFrame";  
export { SyncedEmbedFrame } from "./SyncedEmbedFrame";

REQ-2762: export { EmbedModalContent } from "./EmbedModalContent";

REQ-2763: export { SelectEmbedTypePane } from "./SelectEmbedTypePane";

REQ-2764: export \* from "./PreviewPane";

REQ-2765: import { match } from "ts-pattern";  
  
import type { PreviewBackgroundType } from "./PreviewPane";  
  
export function getCheckerBoardDataUri(  
 theme: Extract<  
 PreviewBackgroundType,  
 ...

REQ-2766: import type { EmbeddingDisplayOptions } from "metabase/public/lib/types";  
  
export function getDefaultDisplayOptions(  
 shouldShownDownloadData: boolean,  
): EmbeddingDisplayOptions {  
 return {  
 ...

REQ-2767: export { StaticEmbedSetupPane } from "./StaticEmbedSetupPane";

REQ-2768: export const EMBED\_MODAL\_TABS = {  
 Overview: "overview" as const,  
 Parameters: "parameters" as const,  
 LookAndFeel: "lookAndFeel" as const,  
};

REQ-2769: import type { EmbedResourceParameter } from "metabase/public/lib/types";  
  
export type ActivePreviewPane = "preview" | "code";  
  
export type EmbedResourceParameterWithValue = EmbedResourceParameter &...

REQ-2770: export function getHighlightedRanges(  
 source: string,  
 highlightedTexts: string[] = [],  
) {  
 return highlightedTexts.flatMap((highlightedText) =>  
 getHighlightedRangesForText(source,...

REQ-2771: import { getHighlightedRanges } from "./utils";  
  
describe("getHighlightedRanges", () => {  
 it("returns an empty array if no highlighted text is provided", () => {  
 ...

REQ-2772: export { EmbedModal } from "./EmbedModal";  
export { EmbedModalContent } from "./EmbedModalContent";

REQ-2773: import { DEFAULT\_DASHBOARD\_DISPLAY\_OPTIONS } from "metabase/dashboard/constants";  
  
export const DEFAULT\_EMBED\_DISPLAY\_PARAMS = {  
 titled: DEFAULT\_DASHBOARD\_DISPLAY\_OPTIONS.titled,  
 theme:...

REQ-2774: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2775: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2776: export { PublicOrEmbeddedDashboardPage } from "./PublicOrEmbeddedDashboardPage";

REQ-2777: export { PublicOrEmbeddedDashboardPage } from "./PublicOrEmbeddedDashboardPage";

REQ-2778: import type { Dashboard } from "metabase-types/api";  
  
export type PublicOrEmbeddedDashboardEventHandlersProps = {  
 /\*\*  
 \* Callback that is called when the dashboard is loaded.  
 \*/  
 onLoad?:...

REQ-2779: import { useCallback, useEffect, useRef } from "react";  
  
/\* eslint-disable-next-line no-restricted-imports -- deprecated sdk import \*/  
import { getEventHandlers } from...

REQ-2780: export { PublicOrEmbeddedQuestion } from "./PublicOrEmbeddedQuestion";

REQ-2781: export { PublicOrEmbeddedQuestion } from "./PublicOrEmbeddedQuestion";

REQ-2782: export \* from "./use-embed-frame-options";  
export \* from "./use-set-embed-font";

REQ-2783: import type { Location } from "history";  
import { useEffect } from "react";  
  
import { useDocsUrl } from "metabase/common/hooks";  
import { parseHashOptions } from "metabase/lib/browser";  
import {...

REQ-2784: import type { Location } from "history";  
import { useEffect } from "react";  
  
import { parseHashOptions } from "metabase/lib/browser";  
import { useDispatch } from "metabase/lib/redux";  
import type {...

REQ-2785: import type { ExportFormatType } from "metabase/embedding/components/PublicLinkPopover/types";  
import { trackSchemaEvent } from "metabase/lib/analytics";  
  
import type {  
 DisplayTheme,  
 ...

REQ-2786: import { match } from "ts-pattern";  
  
import { optionsToHashParams } from "./embed";  
import type {  
 CodeSampleParameters,  
 EmbeddingDisplayOptions,  
 EmbeddingHashOptions,  
 ...

REQ-2787: import {  
 clojure,  
 getHtmlSource,  
 getJsxSource,  
 getPugSource,  
 node,  
 python,  
 ruby,  
} from "./code-templates";  
import type {  
 ClientCodeSampleConfig,  
 CodeSampleParameters,  
 ...

REQ-2788: import { CompactSign } from "jose"; // using jose because jsonwebtoken doesn't work on the web :-/  
import querystring from "querystring";  
  
import type {  
 EmbedResource,  
 EmbedResourceType,  
 ...

REQ-2789: import type { CodeLanguage } from "metabase/common/components/CodeEditor";  
import type { Card, Dashboard } from "metabase-types/api";  
  
export type DisplayTheme = "light" | "night" |...

REQ-2790: import Questions from "metabase/entities/questions";  
import type { Dispatch, GetState } from "metabase-types/store";  
  
// load a card either by ID or from a base64 serialization. if both are present...

REQ-2791: import { createAction } from "redux-actions";  
import \_ from "underscore";  
  
import { invalidateNotificationsApiCache } from "metabase/api";  
import Databases from "metabase/entities/databases";  
import...

REQ-2792: export \* from "./core";  
export \* from "./initializeQB";  
export \* from "./updateQuestion";  
export \* from "./native";

REQ-2793: import type { LocationDescriptorObject } from "history";  
import querystring from "querystring";  
  
import Questions from "metabase/entities/questions";  
import Snippets from...

REQ-2794: import fetchMock from "fetch-mock";  
import type { LocationDescriptorObject } from "history";  
  
import { createMockEntitiesState } from "\_\_support\_\_/store";  
import Databases from...

REQ-2795: export const SET\_IS\_SHOWING\_TEMPLATE\_TAGS\_EDITOR =  
 "metabase/qb/SET\_IS\_SHOWING\_TEMPLATE\_TAGS\_EDITOR";  
export const setIsShowingTemplateTagsEditor = (  
 isShowingTemplateTagsEditor: boolean,  
) =>...

REQ-2796: import { hasMatchingParameters } from "metabase/parameters/utils/dashboards";  
import { setErrorPage } from "metabase/redux/app";  
import { DashboardApi } from "metabase/services";  
import type Metadata...

REQ-2797: import { assocIn } from "icepick";  
import \_ from "underscore";  
  
import \* as Lib from "metabase-lib";  
import type Question from "metabase-lib/v1/Question";  
import type { Series } from...

REQ-2798: export const SOFT\_RELOAD\_CARD = "metabase/qb/SOFT\_RELOAD\_CARD";  
export const API\_UPDATE\_QUESTION = "metabase/qb/API\_UPDATE\_QUESTION";

REQ-2799: import \_ from "underscore";  
  
import { getTrashUndoMessage } from "metabase/archive/utils";  
import Questions from "metabase/entities/questions";  
import { createThunkAction } from...

REQ-2800: import { createMockEntitiesState } from "\_\_support\_\_/store";  
import { checkNotNull } from "metabase/lib/types";  
import \* as questionActions from "metabase/questions/actions";  
import { getMetadata }...

REQ-2801: import \_ from "underscore";  
  
import { syncVizSettingsWithQuery } from "metabase/querying/viz-settings/utils/sync-viz-settings";  
import { getPersistableDefaultSettingsForSeries } from...

REQ-2802: export \* from "./core";  
export \* from "./models";  
export \* from "./native";  
export \* from "./navigation";  
export \* from "./object-detail";  
export \* from "./query-updates";  
export \* from...

REQ-2803: import { createThunkAction } from "metabase/lib/redux";  
import { checkNotNull } from "metabase/lib/types";  
import { UserApi } from "metabase/services";  
  
export const CLOSE\_QB\_NEWB\_MODAL =...

REQ-2804: import { push } from "react-router-redux";  
import { createAction } from "redux-actions";  
import { t } from "ttag";  
  
import { addUndo } from "metabase/redux/undo";  
import type { Dispatch, GetState }...

REQ-2805: import { createAction } from "redux-actions";  
  
import Questions from "metabase/entities/questions";  
import { createThunkAction } from "metabase/lib/redux";  
import { updateUserSetting } from...

REQ-2806: import type { Location } from "history";  
  
import { createThunkAction } from "metabase/lib/redux";  
import { equals } from "metabase/lib/utils";  
import { getLocation } from...

REQ-2807: import \_ from "underscore";  
  
import { createThunkAction } from "metabase/lib/redux";  
import { getMetadata } from "metabase/selectors/metadata";  
import { MetabaseApi } from "metabase/services";  
import...

REQ-2808: import type { Limit } from "metabase-lib";  
import \* as Lib from "metabase-lib";  
import type { Dispatch, GetState } from "metabase-types/store";  
  
import { getQuestion } from "../selectors";  
  
import {...

REQ-2809: import { createAction } from "redux-actions";  
import { t } from "ttag";  
  
import { defer } from "metabase/lib/promise";  
import { createThunkAction } from "metabase/lib/redux";  
import {...

REQ-2810: import { createAction } from "redux-actions";  
  
export const SET\_CURRENT\_STATE = "metabase/qb/SET\_CURRENT\_STATE";  
export const setCurrentState = createAction(SET\_CURRENT\_STATE);

REQ-2811: import { createAction } from "redux-actions";  
  
import type { CollectionId, Timeline } from "metabase-types/api";  
import type { Dispatch, GetState } from "metabase-types/store";  
  
import {...

REQ-2812: import { createAction } from "redux-actions";  
  
import { updateSetting } from "metabase/admin/settings/settings";  
import { getOriginalCard } from "metabase/query\_builder/selectors";  
import {...

REQ-2813: import type { LocationDescriptor } from "history";  
import { push, replace } from "react-router-redux";  
import { parse as parseUrl } from "url";  
  
import { isEqualCard } from...

REQ-2814: import \* as Lib from "metabase-lib";  
import type Question from "metabase-lib/v1/Question";  
import type { VisualizationSettings } from "metabase-types/api";  
import type { Dispatch, GetState } from...

REQ-2815: import type { ObjectId } from "metabase/visualizations/components/ObjectDetail/types";  
import type { Dispatch, GetState } from "metabase-types/store";  
  
import { getPKColumnIndex } from...

REQ-2816: import { trackSchemaEvent, trackSimpleEvent } from "metabase/lib/analytics";  
import \* as Lib from "metabase-lib";  
  
export const trackNewQuestionSaved = (  
 draftQuestion,  
 createdQuestion,  
 ...

REQ-2817: import { useAsync } from "react-use";  
  
import { ActivityApi } from "metabase/services";  
import type { Dashboard } from "metabase-types/api";  
  
export const useMostRecentlyViewedDashboard = () => {  
 ...

REQ-2818: export { AddToDashSelectDashModal } from "./AddToDashSelectDashModal";

REQ-2819: import {  
 coerceCollectionId,  
 isPublicCollection,  
} from "metabase/collections/utils";  
import type { DashboardPickerItem } from "metabase/common/components/Pickers/DashboardPicker";  
import {...

REQ-2820: import userEvent from "@testing-library/user-event";  
  
import { createMockMetadata } from "\_\_support\_\_/metadata";  
import { getIcon, render, renderWithProviders, screen } from "\_\_support\_\_/ui";  
import...

REQ-2821: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2822: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2823: import { createMockEntitiesState } from "\_\_support\_\_/store";  
import { render, renderWithProviders, screen } from "\_\_support\_\_/ui";  
import { checkNotNull } from "metabase/lib/types";  
import {...

REQ-2824: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2825: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2826: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2827: import { render, screen } from "\_\_support\_\_/ui";  
  
import DataSelectorSchemaPicker from "./DataSelectorSchemaPicker";  
  
describe("DataSelectorSchemaPicker", () => {  
 it("displays schema name", () =>...

REQ-2828: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2829: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2830: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2831: import { t } from "ttag";  
  
import type { DataPickerDataType, DataTypeInfoItem } from "./types";  
  
export const CONTAINER\_WIDTH = 300;  
  
export const DATA\_BUCKET: Record<string, DataPickerDataType> = {  
...

REQ-2832: import { t } from "ttag";  
  
export const CARD\_INFO = {  
 question: {  
 get title() {  
 return t`Saved Questions`;  
 },  
 model: "card",  
 icon: "table2",  
 },  
 model: {  
 get title() {  
...

REQ-2833: export const findCollectionById = (collections, collectionId) => {  
 if (!collections || collections.length === 0) {  
 return null;  
 }  
  
 const collection = collections.find((c) => c.id ===...

REQ-2834: import type { IconName } from "metabase/ui";  
  
export type DataPickerDataType =  
 | "models"  
 | "raw-data"  
 | "questions"  
 | "metrics";  
  
export type DataTypeInfoItem = {  
 id: DataPickerDataType;  
 ...

REQ-2835: import {  
 METRICS\_INFO\_ITEM,  
 MODELS\_INFO\_ITEM,  
 RAW\_DATA\_INFO\_ITEM,  
 SAVED\_QUESTIONS\_INFO\_ITEM,  
} from "./constants";  
import type { DataTypeInfoItem } from "./types";  
  
export function...

REQ-2836: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2837: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2838: export \* from "./DatasetNotebook";

REQ-2839: export \* from "./EditorTabs";

REQ-2840: export \* from "./TabHintToast";

REQ-2841: \*

REQ-2842: export { DatasetEditor } from "./DatasetEditor";

REQ-2843: export \* from "./ImpossibleToCreateModelModal";

REQ-2844: import { screen } from "\_\_support\_\_/ui";  
  
import { setup } from "./setup";  
  
describe("ImpossibleToCreateModelModal (OSS)", () => {  
 it("should show a help link when `show-metabase-links: true`", ()...

REQ-2845: import { screen } from "\_\_support\_\_/ui";  
  
import type { SetupOpts } from "./setup";  
import { setup as baseSetup } from "./setup";  
  
function setup(opts: SetupOpts) {  
 baseSetup({...

REQ-2846: import { screen } from "\_\_support\_\_/ui";  
  
import type { SetupOpts } from "./setup";  
import { setup as baseSetup } from "./setup";  
  
function setup(opts: SetupOpts) {  
 baseSetup({  
 ...

REQ-2847: import { json } from "@codemirror/lang-json";  
import {  
 MySQL,  
 PLSQL,  
 PostgreSQL,  
 SQLDialect,  
 StandardSQL,  
 sql,  
} from "@codemirror/lang-sql";  
import {  
 type LanguageSupport,  
 ...

REQ-2848: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2849: export \* from "./NativeQueryEditorActionButtons";

REQ-2850: export \* from "./RightClickPopover";

REQ-2851: export \* from "./VisibilityToggler";

REQ-2852: eslint-disable-next-line import/no-default-export

REQ-2853: export const SCROLL\_MARGIN = 8;  
export const MIN\_HEIGHT\_LINES = 15;  
export const MIN\_EDITOR\_HEIGHT\_AFTER\_DRAGGING = 0;  
export const THRESHOLD\_FOR\_AUTO\_CLOSE = 50;

REQ-2854: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2855: import type { FormatOptionsWithLanguage, SqlLanguage } from "sql-formatter";  
  
import { getEngineNativeType } from "metabase/lib/engine";  
import type NativeQuery from...

REQ-2856: import {  
 canFormatForEngine,  
 formatQuery,  
} from "metabase/query\_builder/components/NativeQueryEditor/utils";  
  
const formattingTestCases = [  
 {  
 engine: "mysql",  
 input:  
 "select...

REQ-2857: export \* from "./NewDatasetModal";

REQ-2858: export { QueryModals } from "./QueryModals";

REQ-2859: import {  
 setupRevisionsEndpoints,  
 setupUsersEndpoints,  
} from "\_\_support\_\_/server-mocks";  
import {  
 renderWithProviders,  
 screen,  
 waitForLoaderToBeRemoved,  
} from "\_\_support\_\_/ui";  
import {...

REQ-2860: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2861: export { QuestionDownloadWidget } from "./QuestionDownloadWidget";

REQ-2862: import { useAsyncFn } from "react-use";  
import type { AsyncFnReturn } from "react-use/lib/useAsyncFn";  
  
import { useDispatch } from "metabase/lib/redux";  
import {  
 type DownloadQueryResultsOpts,  
 ...

REQ-2863: export \* from "./QuestionEmbedWidget";

REQ-2864: import userEvent from "@testing-library/user-event";  
  
import { setupEnterpriseTest } from "\_\_support\_\_/enterprise";  
import { createMockMetadata } from "\_\_support\_\_/metadata";  
import { getIcon,...

REQ-2865: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2866: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2867: export \* from "./AdminEmail";

REQ-2868: export \* from "./AdminEmail";

REQ-2869: export { VisualizationError } from "./VisualizationError";

REQ-2870: import { screen } from "\_\_support\_\_/ui";  
import { createMockCard, createMockDatabase } from "metabase-types/api/mocks";  
  
import { setup } from "./setup";  
  
describe("VisualizationError (OSS)", () =>...

REQ-2871: import { screen } from "\_\_support\_\_/ui";  
import { createMockCard, createMockDatabase } from "metabase-types/api/mocks";  
  
import type { SetupOpts } from "./setup";  
import { setup as baseSetup } from...

REQ-2872: import { screen } from "\_\_support\_\_/ui";  
import { createMockCard, createMockDatabase } from "metabase-types/api/mocks";  
  
import type { SetupOpts } from "./setup";  
import { setup as baseSetup } from...

REQ-2873: import { adjustPositions, stripRemarks } from "../utils";  
  
describe("adjustPositions", () => {  
 const remarkedQuery =  
 "-- Metabase:: userID: 1 queryType: native queryHash:...

REQ-2874: import userEvent from "@testing-library/user-event";  
  
import { render, screen } from "\_\_support\_\_/ui";  
import Warnings from "metabase/query\_builder/components/Warnings";  
  
describe("Warnings", () =>...

REQ-2875: export \* from "./ChartTypeList";

REQ-2876: export \* from "./ChartTypeOption";

REQ-2877: export \* from "./ChartTypeSettings";

REQ-2878: export \* from "./ChartTypeList";  
export \* from "./ChartTypeOption";  
export \* from "./ChartTypeSettings";  
export \* from "./use-question-visualization-state";  
export \* from "./viz-order";

REQ-2879: import { useCallback } from "react";  
import \_ from "underscore";  
  
import visualizations from "metabase/visualizations";  
import { sanatizeResultData } from...

REQ-2880: import type { CardDisplayType } from "metabase-types/api";  
  
export const DEFAULT\_VIZ\_ORDER: CardDisplayType[] = [  
 "table",  
 "bar",  
 "line",  
 "pie",  
 "scalar",  
 "row",  
 "area",  
 "combo",  
 ...

REQ-2881: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2882: import { t } from "ttag";  
  
import { isNotNull } from "metabase/lib/types";  
import \* as Lib from "metabase-lib";  
  
export type ColumnAndSeparator = {  
 separator: string | null;  
 column:...

REQ-2883: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import \* as Lib from "metabase-lib";  
import { columnFinder, createQuery } from "metabase-lib/test-helpers";  
import {  
 createMockDatabase,  
 ...

REQ-2884: export \* from "./CloseModal";  
export \* from "./utils";

REQ-2885: import { useCallback, useEffect, useState } from "react";  
  
import { usePreventPopoverExit } from "metabase/ui/components/utils/PreventPopoverExit";  
  
/\*\*  
 \* useCloseModal sets up click handlers that...

REQ-2886: export \* from "./Errors";

REQ-2887: export const DEBOUNCE\_VALIDATION\_MS = 1000;

REQ-2888: import type { Extension } from "@codemirror/state";  
import { EditorView, tooltips } from "@codemirror/view";  
import { useMemo } from "react";  
  
import { isNotNull } from "metabase/lib/types";  
import {...

REQ-2889: import { LRLanguage, LanguageSupport } from "@codemirror/language";  
import { type Diagnostic, linter } from "@codemirror/lint";  
import type { EditorView } from "@codemirror/view";  
  
import {  
 type...

REQ-2890: import {  
 hasNextSnippetField,  
 hasPrevSnippetField,  
 snippet,  
} from "@codemirror/autocomplete";  
import type { EditorState } from "@codemirror/state";  
import type { EditorView } from...

REQ-2891: export \* from "./ExpressionWidget";  
export \* from "./ExpressionWidgetHeader";

REQ-2892: import \* as Lib from "metabase-lib";  
  
function getNextName(names: string[], name: string, index: number): string {  
 const suffixed = index === 0 ? name : `${name} (${index})`;  
 if...

REQ-2893: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import type \* as Lib from "metabase-lib";  
import { createQuery } from "metabase-lib/test-helpers";  
import {  
 createMockDatabase,  
 ...

REQ-2894: import { t } from "ttag";  
  
import { isNotNull } from "metabase/lib/types";  
import {  
 type HelpText,  
 getHelpText,  
 getSupportedClauses,  
} from "metabase/querying/expressions";  
import \* as Lib from...

REQ-2895: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import { getHelpText } from "metabase/querying/expressions";  
import type \* as Lib from "metabase-lib";  
import { createSampleDatabase } from...

REQ-2896: import { highlight } from "./utils";  
  
// By default css modules dont return classes in unit tests.  
// Since we need the classnames for this test, we mock the...

REQ-2897: import { type Highlighter, type Tag, highlightCode } from "@lezer/highlight";  
  
import { parser } from "metabase/querying/expressions/tokenizer/parser";  
import { classNameForTag } from...

REQ-2898: export \* from "./NameInput";

REQ-2899: import { t } from "ttag";  
  
import type \* as Lib from "metabase-lib";  
  
export function getPlaceholder(expressionMode: Lib.ExpressionMode) {  
 if (expressionMode === "expression") {  
 return t`Give...

REQ-2900: export \* from "./ExpressionWidget";  
export \* from "./CombineColumns";  
export \* from "./ExtractColumn";

REQ-2901: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2902: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2903: export \* from "./SnippetRow";

REQ-2904: export \* from "./SnippetSidebar";

REQ-2905: export { TagEditorHelp } from "./TagEditorHelp";

REQ-2906: import { screen } from "\_\_support\_\_/ui";  
  
import { setup } from "./setup";  
  
describe("TagEditorHelp (OSS)", () => {  
 it("should show a help link when `show-metabase-links: true`", () => {  
 ...

REQ-2907: import { screen } from "\_\_support\_\_/ui";  
  
import type { SetupOpts } from "./setup";  
import { setup as baseSetup } from "./setup";  
  
function setup(opts: SetupOpts) {  
 baseSetup({  
 ...

REQ-2908: import { screen } from "\_\_support\_\_/ui";  
  
import type { SetupOpts } from "./setup";  
import { setup as baseSetup } from "./setup";  
  
function setup(opts: SetupOpts) {  
 baseSetup({  
 ...

REQ-2909: export interface WidgetOption {  
 name?: string;  
 menuName?: string;  
 type: string;  
}

REQ-2910: export \* from "./DataReferenceButton";

REQ-2911: export \* from "./ExecutionTime";

REQ-2912: import { t } from "ttag";  
  
export const formatDuration = (time: number): string => {  
 if (time < 1000) {  
 return t`${time}ms`;  
 }  
  
 return t`${(time / 1000).toFixed(1)}s`;  
};

REQ-2913: import { formatDuration } from "./utils";  
  
describe("formatDuration", () => {  
 it("formats duration correctly", () => {  
 expect(formatDuration(100)).toBe("100ms");  
 ...

REQ-2914: export { NativeCodePanel } from "./NativeCodePanel";

REQ-2915: export { NativeQueryPreview } from "./NativeQueryPreview";

REQ-2916: export \* from "./NativeVariablesButton";

REQ-2917: export \* from "./PreviewQueryButton";

REQ-2918: export { PreviewQueryModal } from "./PreviewQueryModal";

REQ-2919: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2920: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2921: export \* from "./SnippetSidebarButton";

REQ-2922: export { NotebookContainer } from "./NotebookContainer";

REQ-2923: export \* from "./View";

REQ-2924: export \* from "./ViewHeaderContainer";

REQ-2925: export \* from "./ViewLeftSidebarContainer";

REQ-2926: export \* from "./ViewMainContainer";

REQ-2927: export \* from "./ViewNativeQueryEditor";

REQ-2928: export \* from "./ViewRightSidebarContainer";

REQ-2929: export { View } from "./View";

REQ-2930: export \* from "./ViewFooter";

REQ-2931: import userEvent from "@testing-library/user-event";  
import fetchMock from "fetch-mock";  
import { Route } from "react-router";  
import \_ from "underscore";  
  
import { setupGetUserKeyValueEndpoint }...

REQ-2932: export { AdHocQuestionDescription } from "./AdHocQuestionDescription";

REQ-2933: export \* from "./AdHocQuestionLeftSide";

REQ-2934: export \* from "./DashboardBackButton";

REQ-2935: export \* from "./QuestionMoreActionsMenu";

REQ-2936: export { QuestionActions } from "./QuestionActions";

REQ-2937: eslint-disable react/display-name, react/prop-types

REQ-2938: export { QuestionDataSource } from "./QuestionDataSource";

REQ-2939: import { isValidElement } from "react";  
  
import { createMockMetadata } from "\_\_support\_\_/metadata";  
import Question from "metabase-lib/v1/Question";  
import type { Card } from...

REQ-2940: export \* from "./QuestionFiltersHeader";

REQ-2941: export \* from "./QuestionNotebookButton";

REQ-2942: import { useSelector } from "metabase/lib/redux";  
import { getMetadataUnfiltered } from "metabase/selectors/metadata";  
import \* as Lib from "metabase-lib";  
import type Question from...

REQ-2943: export \* from "./SavedQuestionLeftSide";

REQ-2944: export { ToggleNativeQueryPreview } from "./ToggleNativeQueryPreview";

REQ-2945: export \* from "./ViewTitleHeaderRightSide";

REQ-2946: export \* from "./ToggleNativeQueryPreview";  
export \* from "./HeaderBreadcrumbs/HeaderBreadcrumbs";  
export \* from "./QuestionActions";  
export \* from "./QuestionDataSource";  
export \* from...

REQ-2947: export { ViewTitleHeader } from "./ViewTitleHeader";

REQ-2948: import \* as Lib from "metabase-lib";  
import type Question from "metabase-lib/v1/Question";  
  
/\*\*  
 \* We can only "explore results" (i.e. create new questions based on this one)  
 \* when question is a...

REQ-2949: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import Question from "metabase-lib/v1/Question";  
import {  
 createMockCard,  
 createMockNativeDatasetQuery,  
 createMockParameter,  
 ...

REQ-2950: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2951: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import { fireEvent, renderWithProviders, screen } from "\_\_support\_\_/ui";  
import registerVisualizations from...

REQ-2952: export \* from "./ChartTypeSidebar";

REQ-2953: import { render, screen } from "\_\_support\_\_/ui";  
  
import DatasetMetadataStrengthIndicator from "./DatasetMetadataStrengthIndicator";  
  
function setup({ resultMetadata } = {}) {  
 const mockDataset =...

REQ-2954: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2955: export \* from "./ModelCacheManagementSection";

REQ-2956: import { useState } from "react";  
  
export const DEFAULT\_LIST\_LIMIT = 5;  
  
export const useExpandableList = (arr: any[], limit = DEFAULT\_LIST\_LIMIT) => {  
 const [isExpanded, setIsExpanded] =...

REQ-2957: import type { IconData } from "metabase/lib/icon";  
  
export interface QuestionSource {  
 href: string;  
 name: string;  
 model?: string;  
 iconProps?: IconData;  
}

REQ-2958: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import Question from "metabase-lib/v1/Question";  
import { createMockCard } from "metabase-types/api/mocks";  
import {  
 SAMPLE\_DB\_ID,  
 ...

REQ-2959: export { QuestionInfoSidebar } from "./QuestionInfoSidebar";

REQ-2960: export \* from "./QuestionSettingsSidebar";

REQ-2961: export \* from "./AddAggregationButton";

REQ-2962: export \* from "./AggregationItem";

REQ-2963: export \* from "./BreakoutColumnListItem";

REQ-2964: export \* from "./BreakoutColumnList";  
export \* from "./util";  
export \* from "./types";

REQ-2965: import type \* as Lib from "metabase-lib";  
  
export type ListItem = Lib.ColumnDisplayInfo & {  
 column: Lib.ColumnMetadata;  
 breakout?: Lib.BreakoutClause;  
};  
  
export type ListSection = {  
 name:...

REQ-2966: import \* as Lib from "metabase-lib";  
  
import type { ListItem, ListSection } from "./types";  
  
export function getBreakoutListItem(  
 query: Lib.Query,  
 stageIndex: number,  
 breakout:...

REQ-2967: export \* from "./SummarizeAggregationItemList";  
export \* from "./SummarizeBreakoutColumnList";

REQ-2968: export \* from "./SummarizeSidebar";

REQ-2969: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-2970: export const MODAL\_TYPES = {  
 SAVE: "save",  
 ADD\_TO\_DASHBOARD: "add-to-dashboard",  
 MOVE: "move",  
 CLONE: "clone",  
 ARCHIVE: "archive",  
 SAVED: "saved",  
 ADD\_TO\_DASHBOARD\_SAVE:...

REQ-2971: import { useCallback } from "react";  
  
import type { ScheduleCallback } from "metabase/common/hooks/use-callback-effect";  
import { useDispatch } from "metabase/lib/redux";  
import {  
 ...

REQ-2972: import { useCallback } from "react";  
  
import type { ScheduleCallback } from "metabase/common/hooks/use-callback-effect";  
import { useDispatch } from "metabase/lib/redux";  
import type Question from...

REQ-2973: import type {  
 QueryBuilderLoadingControls,  
 QueryBuilderQueryStatus,  
 QueryBuilderUIControls,  
} from "metabase-types/store";  
  
export const DEFAULT\_UI\_CONTROLS: QueryBuilderUIControls = {  
 ...

REQ-2974: export \* from "./use-breakout-query-handlers";  
export \* from "./use-default-query-aggregation";  
export \* from "./types";

REQ-2975: import type \* as Lib from "metabase-lib";  
  
export type UpdateQueryHookProps = {  
 query: Lib.Query;  
 onQueryChange: (nextQuery: Lib.Query) => void;  
 stageIndex: number;  
};

REQ-2976: import { useCallback } from "react";  
  
import \* as Lib from "metabase-lib";  
  
import type { UpdateQueryHookProps } from "./types";  
  
export const useBreakoutQueryHandlers = ({  
 query,  
 onQueryChange,  
...

REQ-2977: import { useCallback, useMemo, useState } from "react";  
  
import \* as Lib from "metabase-lib";  
  
import type { UpdateQueryHookProps } from "./types";  
  
export const useDefaultQueryAggregation = ({  
 ...

REQ-2978: import { useWindowSize } from "react-use";  
  
const INITIAL\_WINDOW\_WIDTH = Infinity;  
  
const BREAKPOINT = 1280;  
  
const NOT\_MOUNTED\_YET = undefined;  
  
type NotMountedYet = typeof NOT\_MOUNTED\_YET;  
  
type...

REQ-2979: import querystring from "querystring";  
import { useEffect, useMemo } from "react";  
  
import { IS\_EMBED\_PREVIEW } from "metabase/lib/embed";  
import { getParameterValuesBySlug } from...

REQ-2980: import { createReducer } from "@reduxjs/toolkit";  
  
import {  
 createCardPublicLink,  
 deleteCardPublicLink,  
 updateCardEmbeddingParams,  
 updateCardEnableEmbedding,  
} from "metabase/api";  
import...

REQ-2981: import { assoc, merge } from "icepick";  
import { handleActions } from "redux-actions";  
import \_ from "underscore";  
  
import {  
 EDIT\_QUESTION,  
 NAVIGATE\_TO\_NEW\_CARD,  
} from...

REQ-2982: import { createSelector } from "@reduxjs/toolkit";  
  
import { getMetadata } from "metabase/selectors/metadata";  
import { getMode as getQuestionMode } from...

REQ-2983: eslint no-use-before-define: "error"

REQ-2984: import { assoc, assocIn } from "icepick";  
  
import { createMockEntitiesState } from "\_\_support\_\_/store";  
import {  
 getIsResultDirty,  
 getIsVisualized,  
 getNativeEditorCursorOffset,  
 ...

REQ-2985: import type { LocationDescriptorObject } from "history";  
  
import type { DatasetEditorTab, QueryBuilderMode } from "metabase-types/store";  
  
type LocationQBModeResult = {  
 queryBuilderMode:...

REQ-2986: import \* as Lib from "metabase-lib";  
  
import type { UpdateQueryHookProps } from "../hooks/types";  
  
export type AggregationItem = {  
 aggregation: Lib.AggregationClause;  
 aggregationIndex: number;  
 ...

REQ-2987: import type { Location } from "history";  
import querystring from "querystring";  
import \_ from "underscore";  
  
import { serializeCardForUrl } from "metabase/lib/card";  
import \* as Urls from...

REQ-2988: import \* as Lib from "metabase-lib";  
import type Question from "metabase-lib/v1/Question";  
import { isAdHocModelOrMetricQuestion } from "metabase-lib/v1/metadata/utils/models";  
  
export function...

REQ-2989: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import { getNextId } from "\_\_support\_\_/utils";  
import { serializeCardForUrl } from "metabase/lib/card";  
import { checkNotNull } from...

REQ-2990: import { trackSchemaEvent } from "metabase/lib/analytics";  
import \* as Lib from "metabase-lib";  
import type Question from "metabase-lib/v1/Question";  
  
export const trackColumnCombineViaColumnHeader =...

REQ-2991: import type { Drill } from "metabase/visualizations/types";  
import type \* as Lib from "metabase-lib";  
  
import { automaticInsightsDrill } from "./automatic-insights-drill";  
import { columnExtractDrill...

REQ-2992: import { t } from "ttag";  
  
import type { Drill } from "metabase/visualizations/types/click-actions";  
import type \* as Lib from "metabase-lib";  
  
export const distributionDrill:...

REQ-2993: import { t } from "ttag";  
  
import type { Drill } from "metabase/visualizations/types/click-actions";  
import type \* as Lib from "metabase-lib";  
  
export const fkDetailsDrill:...

REQ-2994: import dayjs from "dayjs";  
import { t } from "ttag";  
  
import {  
 pluralize,  
 singularize,  
 stripId,  
} from "metabase/lib/formatting/strings";  
import type { Drill } from...

REQ-2995: import { t } from "ttag";  
  
import type { Drill } from "metabase/visualizations/types/click-actions";  
import type \* as Lib from "metabase-lib";  
  
export const pkDrill: Drill<Lib.PKDrillThruInfo> = ({  
 ...

REQ-2996: import { isNotNull } from "metabase/lib/types";  
import type { ClickAction } from "metabase/visualizations/types";  
import type { DrillThruDisplayInfo } from "metabase-lib";  
import \* as Lib from...

REQ-2997: import { t } from "ttag";  
  
import type {  
 ClickActionBase,  
 Drill,  
} from "metabase/visualizations/types/click-actions";  
import type \* as Lib from "metabase-lib";  
  
const ACTIONS: Record<string,...

REQ-2998: import { t } from "ttag";  
  
import type { Drill } from "metabase/visualizations/types/click-actions";  
import type \* as Lib from "metabase-lib";  
  
export const summarizeColumnByTimeDrill: Drill<  
 ...

REQ-2999: import { t } from "ttag";  
  
import type {  
 ClickActionBase,  
 Drill,  
} from "metabase/visualizations/types/click-actions";  
import type \* as Lib from "metabase-lib";  
import type { Dispatch } from...

REQ-3000: import dayjs from "dayjs";  
import { msgid, ngettext } from "ttag";  
  
import { inflect } from "metabase/lib/formatting/strings";  
import type {  
 Drill,  
 QuestionChangeClickAction,  
} from...

REQ-3001: import { t } from "ttag";  
  
import { zoomInRow } from "metabase/query\_builder/actions";  
import type { Drill } from "metabase/visualizations/types/click-actions";  
import type \* as Lib from...

REQ-3002: import { t } from "ttag";  
  
import type { Drill } from "metabase/visualizations/types";  
import type \* as Lib from "metabase-lib";  
  
export const zoomInBinningDrill: Drill<Lib.ZoomDrillThruInfo> = ({  
 ...

REQ-3003: import { t } from "ttag";  
  
import type { Drill } from "metabase/visualizations/types";  
import type \* as Lib from "metabase-lib";  
  
export const zoomInGeographicDrill: Drill<Lib.ZoomDrillThruInfo> =...

REQ-3004: import type { Drill } from "metabase/visualizations/types";  
import type \* as Lib from "metabase-lib";  
  
export const zoomInTimeseriesDrill: Drill<Lib.ZoomTimeseriesDrillThruInfo> = ({  
 drill,  
 ...

REQ-3005: import \_ from "underscore";  
  
import { isNotNull } from "metabase/lib/types";  
import type \* as Lib from "metabase-lib";  
import type Database from "metabase-lib/v1/metadata/Database";  
  
import {  
 ...

REQ-3006: import \* as Lib from "metabase-lib";  
  
import { type ExpressionError, renderError } from "./errors";  
import { compile, lexify, parse } from "./pratt";  
import { type Resolver, resolver as...

REQ-3007: import \* as Lib from "metabase-lib";  
  
import { compileExpression } from "./compile-expression";  
import {  
 expressions,  
 fields,  
 query,  
 segments,  
 stageIndex,  
} from "./test/shared";  
import {...

REQ-3008: import dayjs from "dayjs";  
import { t } from "ttag";  
  
import type \* as Lib from "metabase-lib";  
  
import { defineClauses, dimension, op } from "./define";  
import { MBQLClauseCategory as CATEGORY }...

REQ-3009: import type { MBQLClauseDefinition } from "./types";  
  
const names = new Set();  
  
export function defineClauses<  
 const T extends Record<string, MBQLClauseDefinition>,  
>(  
 options:...

REQ-3010: import type \* as Lib from "metabase-lib";  
import type Metadata from "metabase-lib/v1/metadata/Metadata";  
  
import { compileExpression } from "../compile-expression";  
import type { ExpressionError }...

REQ-3011: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import \* as Lib from "metabase-lib";  
import type Metadata from "metabase-lib/v1/metadata/Metadata";  
import { createSampleDatabase } from...

REQ-3012: import { msgid, ngettext } from "ttag";  
  
import \* as Lib from "metabase-lib";  
  
import { getClauseDefinition } from "../../clause";  
import { visit } from "../../visitor";  
import { error } from...

REQ-3013: import \* as Lib from "metabase-lib";  
  
import { getClauseDefinition } from "../../clause";  
import { visit } from "../../visitor";  
import { error } from "../utils";  
  
export function...

REQ-3014: import { t } from "ttag";  
  
import \* as Lib from "metabase-lib";  
  
import { visit } from "../../visitor";  
import { error } from "../utils";  
  
export function checkCaseOrIfArgCount({  
 ...

REQ-3015: import { t } from "ttag";  
  
import \* as Lib from "metabase-lib";  
  
import { COMPARISON, EQUALITY } from "../../pratt";  
import { parsePunctuator } from "../../punctuator";  
import { visit } from...

REQ-3016: import \* as Lib from "metabase-lib";  
  
import { checkExpressionModeSupportsClause } from "../../mode";  
import { visit } from "../../visitor";  
  
export function checkFunctionsForExpressionMode({  
 ...

REQ-3017: import { t } from "ttag";  
  
import \* as Lib from "metabase-lib";  
  
import { getClauseDefinition } from "../../clause";  
import { visit } from "../../visitor";  
import { error } from "../utils";  
  
export...

REQ-3018: import \* as Lib from "metabase-lib";  
  
import { DiagnosticError } from "../../errors";  
  
export function checkLibDiagnostics({  
 query,  
 stageIndex,  
 expressionMode,  
 expressionClause,  
 ...

REQ-3019: import { t } from "ttag";  
  
import \* as Lib from "metabase-lib";  
import type Metadata from "metabase-lib/v1/metadata/Metadata";  
  
import { getClauseDefinition } from "../../clause";  
import {...

REQ-3020: import type \* as Lib from "metabase-lib";  
import type Metadata from "metabase-lib/v1/metadata/Metadata";  
  
import { checkArgCount } from "./check-arg-count";  
import { checkArgValidators } from...

REQ-3021: export \* from "./diagnostics";

REQ-3022: import { t } from "ttag";  
  
import { BAD\_TOKEN, type Token } from "../../pratt";  
import { error } from "../utils";  
  
export function checkBadTokens({ tokens }: { tokens: Token[] }) {  
 for (const token...

REQ-3023: import { t } from "ttag";  
  
import { COMMA, type Token } from "../../pratt";  
import { error } from "../utils";  
  
export function checkDoubleCommas({ tokens }: { tokens: Token[] }) {  
 for (let i = 1; i...

REQ-3024: import { t } from "ttag";  
  
import { FIELD, type Token } from "../../pratt";  
import { quoteString } from "../../string";  
import { error } from "../utils";  
  
export function checkFieldQuotes({ tokens }:...

REQ-3025: import { t } from "ttag";  
  
import type { Token } from "../../pratt";  
import { GROUP, GROUP\_CLOSE } from "../../pratt";  
import { error } from "../utils";  
  
export function checkMatchingParentheses({...

REQ-3026: import { t } from "ttag";  
  
import {  
 BAD\_TOKEN,  
 BOOLEAN,  
 CALL,  
 END\_OF\_INPUT,  
 FIELD,  
 GROUP,  
 GROUP\_CLOSE,  
 IDENTIFIER,  
 NUMBER,  
 STRING,  
 type Token,  
} from "../../pratt";  
import {...

REQ-3027: import { t } from "ttag";  
  
import { NUMBER, type Token } from "../../pratt";  
import { error } from "../utils";  
  
export function checkNumberExponent({ tokens }: { tokens: Token[] }) {  
 for (const...

REQ-3028: import { t } from "ttag";  
  
import { getClauseDefinition, getMBQLName } from "../../clause";  
import { GROUP, IDENTIFIER, type Token } from "../../pratt";  
import { error } from "../utils";  
  
export...

REQ-3029: import { t } from "ttag";  
  
import { STRING, type Token } from "../../pratt";  
import { quoteString } from "../../string";  
import { error } from "../utils";  
  
export function checkStringQuotes({ tokens...

REQ-3030: import type { Token } from "../../pratt";  
  
import { checkBadTokens } from "./check-bad-tokens";  
import { checkDoubleCommas } from "./check-double-commas";  
import { checkFieldQuotes } from...

REQ-3031: import type \* as Lib from "metabase-lib";  
  
import { DiagnosticError } from "../errors";  
import { type Node, Token } from "../pratt";  
  
export function error(message: string): never;  
export function...

REQ-3032: import { CALL, Token } from "../pratt";  
  
import { position } from "./utils";  
  
describe("position", () => {  
 it("returns the correct position", () => {  
 const token = new Token({  
 type:...

REQ-3033: import { t } from "ttag";  
  
import type { Node } from "./pratt";  
  
/\*  
 \* This class helps anything that handles parser errors to use instanceof to  
 \* easily distinguish between compilation error...

REQ-3034: import type { AstPath, Doc, ParserOptions, Plugin } from "prettier";  
import { builders } from "prettier/doc";  
import { format as pformat } from "prettier/standalone";  
  
import { parseNumber } from...

REQ-3035: eslint-disable jest/expect-expect

REQ-3036: export \* from "./formatter";

REQ-3037: import type { AstPath } from "prettier";  
  
import \* as Lib from "metabase-lib";  
  
import { EXPRESSION\_OPERATORS } from "../config";  
import \* as literal from "../literal";  
  
type Assertion<T> = T extends...

REQ-3038: import \* as Lib from "metabase-lib";  
  
import { compileExpression } from "./compile-expression";  
import { fuzz } from "./test/fuzz";  
import { generateExpression } from "./test/generator";  
import {...

REQ-3039: import { quoteString, unquoteString } from "./string";  
import { fuzz } from "./test/fuzz";  
import { createRandom } from "./test/generator";  
  
const MAX\_SEED = 1000;  
  
const simple = [  
 ...

REQ-3040: import { isNotNull } from "metabase/lib/types";  
import type \* as Lib from "metabase-lib";  
import type Database from "metabase-lib/v1/metadata/Database";  
  
import { getClauseDefinition } from...

REQ-3041: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import { checkNotNull } from "metabase/lib/types";  
import type { Database } from "metabase-types/api";  
import { createMockDatabase } from...

REQ-3042: import { FK\_SYMBOL } from "metabase/lib/formatting/constants";  
  
import { quoteString } from "./string";  
  
const IDENTIFIER\_QUOTE = "[";  
  
export const EDITOR\_FK\_SYMBOLS = {  
 // specifies which symbols...

REQ-3043: export \* from "./clause";  
export \* from "./diagnostics";  
export \* from "./errors";  
export \* from "./formatter";  
export \* from "./help-text";  
export \* from "./mode";  
export \* from "./position";  
export...

REQ-3044: import type {  
 BooleanLiteral,  
 NumericLiteral,  
 StringLiteral,  
} from "metabase-types/api";  
  
export function isLiteral(  
 x: unknown,  
): x is StringLiteral | NumericLiteral | BooleanLiteral {  
 ...

REQ-3045: import { t } from "ttag";  
  
import \* as Lib from "metabase-lib";  
  
import { getClauseDefinition, isDefinedClause } from "./clause";  
import { DiagnosticError } from "./errors";  
  
const...

REQ-3046: import { getMBQLName } from "./clause";  
import { END\_OF\_INPUT, STRING, type Token, lexify } from "./pratt";  
import { parser } from "./tokenizer/parser";  
  
export function enclosingFunction(doc:...

REQ-3047: import expression from "ts-dedent";  
  
import { enclosingFunction } from "./position";  
  
describe("enclosingFunction", () => {  
 function setup(doc: string) {  
 const pos = doc.indexOf("|");  
 if...

REQ-3048: import { t } from "ttag";  
  
import { type NumberValue, parseNumber } from "metabase/lib/number";  
import \* as Lib from "metabase-lib";  
  
import { getClauseDefinition, getMBQLName, isDefinedClause } from...

REQ-3049: import \* as Lib from "metabase-lib";  
  
import { compileExpression } from "../compile-expression";  
import { query } from "../test/shared";  
import { value } from "../test/utils";  
import type {...

REQ-3050: import { lexify } from "../pratt/lexifier";  
import { fuzz } from "../test/fuzz";  
import { generateExpression } from "../test/generator";  
  
describe("metabase/querying/expressions/tokenizer", () => {  
 ...

REQ-3051: export \* from "./lexifier";  
export \* from "./parser";  
export \* from "./compiler";  
export \* from "./syntax";  
export \* from "./node";  
export \* from "./token";

REQ-3052: import type { SyntaxNodeRef } from "@lezer/common";  
  
import { parsePunctuator } from "../punctuator";  
import { unquoteString } from "../string";  
import { tokenize } from "../tokenizer";  
  
import type...

REQ-3053: import { lexify } from "./lexifier";  
import type { NodeType } from "./node";  
import {  
 ADD,  
 BAD\_TOKEN,  
 BOOLEAN,  
 CALL,  
 COMMA,  
 COMPARISON,  
 END\_OF\_INPUT,  
 FIELD,  
 GROUP,  
 GROUP\_CLOSE,  
 ...

REQ-3054: import type { Token } from "./token";  
  
export type NodeType = {  
 name?: string;  
  
 // Number of operands to expect for this node on the left side  
 leftOperands: number;  
 // Number of operands to...

REQ-3055: import { t } from "ttag";  
  
import { CompileError } from "../errors";  
import type { Hooks } from "../types";  
import { assert } from "../utils";  
  
import type { Node, NodeType } from "./node";  
import {  
...

REQ-3056: import { lexify, parse } from ".";  
  
describe("pratt/parser", () => {  
 function parseExpression(source: string) {  
 const tokens = lexify(source);  
 return parse(tokens, {  
 hooks: {  
 ...

REQ-3057: import type { Node, NodeType } from "./node";  
  
/\*  
 \* This file specifies most of the syntax for the Metabase handwritten custom  
 \* expression parser. The rest is contained in the parser special...

REQ-3058: import type { NodeType } from "./node";  
  
export class Token {  
 type: NodeType;  
 text: string;  
 value?: string;  
  
 start: number;  
 end: number;  
  
 constructor({  
 type,  
 start,  
 end,  
 ...

REQ-3059: import type { NodeType } from "./pratt";  
import { NODE\_TYPE as t } from "./pratt/syntax";  
  
export type Punctuator = keyof typeof PUNCTUATOR\_TO\_TYPE;  
  
const PUNCTUATOR\_TO\_TYPE = {  
 ",": t.COMMA,  
 ...

REQ-3060: import { c, t } from "ttag";  
import \_ from "underscore";  
  
import \* as Lib from "metabase-lib";  
  
import { CompileError } from "./errors";  
import { EDITOR\_FK\_SYMBOLS, getDisplayNameWithSeparator } from...

REQ-3061: import \* as Lib from "metabase-lib";  
  
import { columnsForExpressionMode } from "./mode";  
import { resolver as makeResolver } from "./resolver";  
import {  
 expressions,  
 fields,  
 findDimensions,  
 ...

REQ-3062: const DOUBLE\_QUOTE = '"';  
const SINGLE\_QUOTE = "'";  
const OPEN\_BRACKET = "[";  
const CLOSE\_BRACKET = "]";  
const BACKSLASH = "\\";  
  
export const STRING\_LITERAL\_DEFAULT\_QUOTE = '"';  
  
export type...

REQ-3063: import { quoteString, unquoteString } from "./string";  
  
const dq = (str: string) => quoteString(str, '"');  
const sq = (str: string) => quoteString(str, "'");  
const bq = (str: string) =>...

REQ-3064: import {  
 CompletionContext,  
 type CompletionSource,  
} from "@codemirror/autocomplete";  
import { EditorState } from "@codemirror/state";  
  
export function complete(source: CompletionSource | null,...

REQ-3065: import type { CompletionContext } from "@codemirror/autocomplete";  
  
import type \* as Lib from "metabase-lib";  
import type Metadata from "metabase-lib/v1/metadata/Metadata";  
  
import {...

REQ-3066: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import { createQuery } from "metabase-lib/test-helpers";  
import type { DatabaseFeature } from "metabase-types/api";  
import {...

REQ-3067: import type { CompletionContext } from "@codemirror/autocomplete";  
  
import { getColumnIcon } from "metabase/common/utils/columns";  
import \* as Lib from "metabase-lib";  
  
import { formatIdentifier }...

REQ-3068: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import \* as Lib from "metabase-lib";  
import { SAMPLE\_DATABASE, createQuery } from "metabase-lib/test-helpers";  
import type { DatasetQuery,...

REQ-3069: import type { CompletionContext } from "@codemirror/autocomplete";  
  
import type \* as Lib from "metabase-lib";  
import type Metadata from "metabase-lib/v1/metadata/Metadata";  
  
import {...

REQ-3070: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import { createQuery } from "metabase-lib/test-helpers";  
import type { DatabaseFeature } from "metabase-types/api";  
import {...

REQ-3071: export \* from "./suggest";  
export \* from "./util";  
export { type ExpressionSuggestion } from "./types";

REQ-3072: import type { CompletionContext } from "@codemirror/autocomplete";  
  
import { tokenAtPos } from "../position";  
  
import { isFieldReference, isIdentifier } from "./util";  
  
export function...

REQ-3073: import { complete } from "./\_\_support\_\_";  
import { suggestLiterals } from "./literals";  
  
describe("suggestLiterals", () => {  
 it("should suggest True and False", () => {  
 const results =...

REQ-3074: import type { CompletionContext } from "@codemirror/autocomplete";  
  
import \* as Lib from "metabase-lib";  
  
import { formatIdentifier } from "../identifier";  
import { tokenAtPos } from...

REQ-3075: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import { SAMPLE\_DATABASE, createQuery } from "metabase-lib/test-helpers";  
import {  
 createMockCard,  
 createMockField,  
 ...

REQ-3076: import type { CompletionContext } from "@codemirror/autocomplete";  
  
import \* as Lib from "metabase-lib";  
  
import { formatIdentifier } from "../identifier";  
import { tokenAtPos } from...

REQ-3077: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import { SAMPLE\_DATABASE, createQuery } from "metabase-lib/test-helpers";  
import { createMockSegment, createMockTable } from...

REQ-3078: import { autocompletion } from "@codemirror/autocomplete";  
  
import { isNotNull } from "metabase/lib/types";  
import type \* as Lib from "metabase-lib";  
import type Metadata from...

REQ-3079: import type {  
 Completion as CodeMirrorCompletion,  
 CompletionResult as CodeMirrorCompletionResult,  
} from "@codemirror/autocomplete";  
  
import type { IconName } from "metabase/ui";  
import type \* as...

REQ-3080: import { snippetCompletion } from "@codemirror/autocomplete";  
import Fuse from "fuse.js";  
  
import { CALL, FIELD, IDENTIFIER, type Token } from "../pratt";  
import type { MBQLClauseFunctionConfig }...

REQ-3081: import \_ from "underscore";  
  
export const fuzz = process.env.MB\_FUZZ ? describe : \_.noop;

REQ-3082: type Generator<T> = () => T;  
  
type Node = {  
 type: number;  
 value?: number | string;  
 params?: Node[];  
 left?: Node;  
 right?: Node;  
 child?: Node;  
 op?: string;  
};  
  
function assert<T>(x: T |...

REQ-3083: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import { getNextId } from "\_\_support\_\_/utils";  
import \* as Lib from "metabase-lib";  
import {  
 type ExpressionClauseOpts,  
 createQuery,  
 ...

REQ-3084: import type \* as Lib from "metabase-lib";  
  
export function op(  
 operator: string,  
 ...args: (Lib.ExpressionParts | Lib.ExpressionArg)[]  
): Lib.ExpressionParts {  
 return {  
 operator: operator as...

REQ-3085: export \* from "./tokenize";

REQ-3086: This file was generated by lezer-generator. You probably shouldn't edit it.

REQ-3087: This file was generated by lezer-generator. You probably shouldn't edit it.

REQ-3088: import { styleTags, tags as t } from "@lezer/highlight";  
  
import { parser as baseParser } from "./lezer";  
  
export const tags = styleTags({  
 Identifier: t.variableName,  
 Boolean: t.bool,  
 True:...

REQ-3089: import { parser } from "./parser";  
  
export function tokenize(source: string) {  
 const tree = parser.parse(source);  
 return tree.cursor();  
}

REQ-3090: import { ExternalTokenizer } from "@lezer/lr";  
  
import { Field } from "./lezer.terms";  
  
function char(char: string): number {  
 if (char.length !== 1) {  
 throw new Error(`Expected a single...

REQ-3091: import type \* as Lib from "metabase-lib";  
import type Database from "metabase-lib/v1/metadata/Database";  
import type { DatabaseFeature } from "metabase-types/api";  
  
import type { DefinedClauseName }...

REQ-3092: import \* as Lib from "metabase-lib";  
import type Database from "metabase-lib/v1/metadata/Database";  
import type Metadata from "metabase-lib/v1/metadata/Metadata";  
  
import { CompileError } from...

REQ-3093: import \* as Lib from "metabase-lib";  
  
type Visitor = (expression: Lib.ExpressionParts | Lib.ExpressionArg) => void;  
  
export function visit(  
 node: Lib.ExpressionParts | Lib.ExpressionArg,  
 visitor:...

REQ-3094: export \* from "./FieldPanel";

REQ-3095: import type \* as Lib from "metabase-lib";  
  
export type ColumnItem = {  
 column: Lib.ColumnMetadata;  
 displayName: string;  
 isSelected: boolean;  
 isDisabled: boolean;  
 isHidden?:...

REQ-3096: import \* as Lib from "metabase-lib";  
  
import type { ColumnGroupItem, ColumnItem } from "./types";  
  
function getColumns(query: Lib.Query, stageIndex: number) {  
 const aggregations =...

REQ-3097: export \* from "./BooleanPicker";

REQ-3098: import { t } from "ttag";  
  
import type { OperatorOption } from "./types";  
  
export const OPERATOR\_OPTIONS: OperatorOption[] = [  
 {  
 get label() {  
 return t`All time`;  
 },  
 value:...

REQ-3099: export \* from "./DateOperatorPicker";

REQ-3100: import type {  
 DatePickerOperator,  
 RelativeIntervalDirection,  
} from "metabase/querying/filters/types";  
  
export type OptionType =  
 | "none"  
 | DatePickerOperator  
 |...

REQ-3101: import type {  
 DatePickerOperator,  
 DatePickerValue,  
} from "metabase/querying/filters/types";  
  
import { getExcludeOperatorValue } from "../ExcludeDatePicker/utils";  
import {  
 ...

REQ-3102: import { DATE\_PICKER\_OPERATORS } from "metabase/querying/filters/constants";  
import type {  
 DatePickerTruncationUnit,  
 DatePickerValue,  
 ExcludeDatePickerValue,  
 RelativeDatePickerValue,  
 ...

REQ-3103: import { t } from "ttag";  
  
import type { ShortcutOption } from "metabase/querying/filters/types";  
  
import type { TypeOption } from "./types";  
  
const DAY\_WEEK\_SHORTCUT\_OPTIONS: ShortcutOption[] = [  
 ...

REQ-3104: export \* from "./DateShortcutPicker";  
export { getShortcutOptions } from "./utils";

REQ-3105: import type {  
 DatePickerOperator,  
 DatePickerValueType,  
} from "metabase/querying/filters/types";  
  
export interface TypeOption {  
 label: string;  
 type: DatePickerValueType;  
 operators:...

REQ-3106: import type {  
 DatePickerOperator,  
 DatePickerShortcut,  
 ShortcutOption,  
} from "metabase/querying/filters/types";  
  
import { SHORTCUT\_OPTION\_GROUPS, TYPE\_OPTIONS } from "./constants";  
import type...

REQ-3107: import { t } from "ttag";  
  
import type { ExcludeOperatorOption, ExcludeUnitOption } from "./types";  
  
export const EXCLUDE\_UNIT\_OPTIONS: ExcludeUnitOption[] = [  
 {  
 unit: "day-of-week",  
 get...

REQ-3108: export \* from "./ExcludeDatePicker";

REQ-3109: import type {  
 DatePickerExtractionUnit,  
 ExcludeDatePickerOperator,  
} from "metabase/querying/filters/types";  
  
export interface ExcludeUnitOption {  
 unit: DatePickerExtractionUnit;  
 label:...

REQ-3110: import dayjs from "dayjs";  
import { t } from "ttag";  
import \_ from "underscore";  
  
import type {  
 DatePickerExtractionUnit,  
 DatePickerOperator,  
 DatePickerUnit,  
 ExcludeDatePickerOperator,  
 ...

REQ-3111: import type { DatePickerTruncationUnit } from "metabase/querying/filters/types";  
  
export const UNIT\_GROUPS: DatePickerTruncationUnit[][] = [  
 ["day", "week", "month"],  
 ["quarter", "year"],  
];

REQ-3112: export \* from "./CurrentDatePicker";

REQ-3113: import type {  
 DatePickerTruncationUnit,  
 DatePickerUnit,  
 RelativeDatePickerValue,  
} from "metabase/querying/filters/types";  
  
import { UNIT\_GROUPS } from "./constants";  
  
export function...

REQ-3114: export \* from "./SimpleDateIntervalPicker";

REQ-3115: import type { DatePickerTruncationUnit } from "metabase/querying/filters/types";  
  
export const DEFAULT\_OFFSETS: Record<DatePickerTruncationUnit, number> = {  
 minute: 60,  
 hour: 24,  
 day: 7,  
 ...

REQ-3116: export \* from "./DateIntervalPicker";

REQ-3117: import type {  
 DatePickerTruncationUnit,  
 RelativeDatePickerValue,  
} from "metabase/querying/filters/types";  
import \* as Lib from "metabase-lib";  
  
import { DEFAULT\_OFFSETS } from...

REQ-3118: export \* from "./DateOffsetIntervalPicker";

REQ-3119: import { t } from "ttag";  
  
import type {  
 DatePickerTruncationUnit,  
 DatePickerUnit,  
 RelativeDatePickerValue,  
 RelativeIntervalDirection,  
} from "metabase/querying/filters/types";  
import \* as...

REQ-3120: export \* from "./IncludeCurrentSwitch";

REQ-3121: import { t } from "ttag";  
  
import type { Tab } from "./types";  
  
export const TABS: Tab[] = [  
 {  
 get label() {  
 return t`Previous`;  
 },  
 direction: "last",  
 },  
 {  
 get label() {  
 ...

REQ-3122: export \* from "./RelativeDatePicker";

REQ-3123: import type { RelativeIntervalDirection } from "metabase/querying/filters/types";  
  
export interface Tab {  
 label: string;  
 direction: RelativeIntervalDirection;  
}

REQ-3124: import { DATE\_PICKER\_TRUNCATION\_UNITS } from "metabase/querying/filters/constants";  
import type {  
 DatePickerTruncationUnit,  
 DatePickerUnit,  
 DatePickerValue,  
 RelativeDatePickerValue,  
 ...

REQ-3125: import type {  
 DatePickerTruncationUnit,  
 RelativeDatePickerValue,  
} from "metabase/querying/filters/types";  
  
import { setDirection } from "./utils";  
  
describe("setDirection", () => {  
 ...

REQ-3126: export \* from "./SimpleDatePicker";

REQ-3127: export \* from "./DateRangePickerBody";

REQ-3128: export \* from "./SimpleDateRangePicker";

REQ-3129: export \* from "./DateRangePicker";  
export \* from "./types";

REQ-3130: export interface DateRangePickerValue {  
 dateRange: [Date, Date];  
 hasTime: boolean;  
}

REQ-3131: export \* from "./SimpleSpecificDatePicker";

REQ-3132: export \* from "./SimpleSingleDatePicker";

REQ-3133: export \* from "./SingleDatePickerBody";

REQ-3134: export \* from "./SingleDatePicker";  
export \* from "./types";

REQ-3135: export interface SingleDatePickerValue {  
 date: Date;  
 hasTime: boolean;  
}

REQ-3136: export \* from "./TimeToggle";

REQ-3137: import { t } from "ttag";  
  
import type { Tab } from "./types";  
  
export const TABS: Tab[] = [  
 {  
 get label() {  
 return t`Between`;  
 },  
 operator: "between",  
 },  
 {  
 get label() {  
...

REQ-3138: export \* from "./SpecificDatePicker";

REQ-3139: import type { SpecificDatePickerOperator } from "metabase/querying/filters/types";  
  
export interface Tab {  
 label: string;  
 operator: SpecificDatePickerOperator;  
}

REQ-3140: import dayjs from "dayjs";  
  
import type {  
 DatePickerOperator,  
 DatePickerUnit,  
 DatePickerValue,  
 SpecificDatePickerOperator,  
 SpecificDatePickerValue,  
} from...

REQ-3141: export const MIN\_WIDTH = 300;

REQ-3142: export \* from "./DatePicker";

REQ-3143: import type { DatePickerValue } from "metabase/querying/filters/types";  
  
export type DatePickerSubmitButtonProps = {  
 value: DatePickerValue;  
 isDisabled?: boolean;  
};

REQ-3144: export \* from "./FilterPanelPopover";

REQ-3145: export \* from "./FilterPill";

REQ-3146: export \* from "./FilterPanel";

REQ-3147: import type \* as Lib from "metabase-lib";  
  
export interface FilterItem {  
 filter: Lib.FilterClause;  
 filterIndex: number;  
 stageIndex: number;  
}

REQ-3148: import \* as Lib from "metabase-lib";  
  
import type { FilterItem } from "./types";  
  
export function getFilterItems(query: Lib.Query): FilterItem[] {  
 const stageIndexes = Lib.stageIndexes(query);  
 ...

REQ-3149: import \* as Lib from "metabase-lib";  
import { createQuery } from "metabase-lib/test-helpers";  
  
import { getFilterItems } from "./utils";  
  
const STAGE\_COUNT = 4;  
  
function createFilteredQuery(query:...

REQ-3150: export \* from "./BooleanFilterPicker";

REQ-3151: export \* from "./CoordinateColumnPicker";

REQ-3152: import type \* as Lib from "metabase-lib";  
  
export interface ColumnOption {  
 value: string;  
 label: string;  
 column: Lib.ColumnMetadata;  
}

REQ-3153: import { t } from "ttag";  
  
import \* as Lib from "metabase-lib";  
  
import type { ColumnOption } from "./types";  
  
export function getColumnOptions(  
 query: Lib.Query,  
 stageIndex: number,  
 columns:...

REQ-3154: export \* from "./CoordinateFilterPicker";

REQ-3155: export \* from "./SimpleDateFilterPicker";

REQ-3156: export \* from "./DateFilterPicker";

REQ-3157: export \* from "./DefaultFilterPicker";

REQ-3158: export \* from "./FilterColumnPicker";

REQ-3159: export \* from "./FilterOperatorPicker";

REQ-3160: export \* from "./FilterPickerBody";

REQ-3161: export \* from "./FilterPickerFooter";

REQ-3162: export \* from "./FilterPickerHeader";

REQ-3163: export \* from "./FilterSubmitButton";

REQ-3164: export \* from "./MultiStageFilterPicker";

REQ-3165: export \* from "./NumberFilterPicker";

REQ-3166: export \* from "./StringFilterPicker";

REQ-3167: export \* from "./TimeFilterPicker";

REQ-3168: import type { ComboboxProps } from "metabase/ui";  
  
export const WIDTH = 380;  
export const COMBOBOX\_PROPS: Partial<ComboboxProps> = {  
 width: 298,  
 position: "bottom-start",  
};

REQ-3169: export { FilterPicker, type FilterPickerProps } from "./FilterPicker";

REQ-3170: istanbul ignore file

REQ-3171: import type { DefinedClauseName } from "metabase/querying/expressions";  
import type \* as Lib from "metabase-lib";  
  
export type FilterPickerWidgetProps = {  
 autoFocus: boolean;  
 query: Lib.Query;  
 ...

REQ-3172: export \* from "./ListValuePicker";

REQ-3173: import type { ComboboxItem } from "metabase/ui";  
import type { FieldValue } from "metabase-types/api";  
  
import { getFieldOptions } from "../utils";  
  
export function searchOptions(  
 options:...

REQ-3174: export const SEARCH\_DEBOUNCE = 500;  
export const SEARCH\_LIMIT = 100;

REQ-3175: export \* from "./SearchValuePicker";

REQ-3176: import { t } from "ttag";  
  
import type { FieldValue } from "metabase-types/api";  
  
import { SEARCH\_LIMIT } from "./constants";  
  
export function shouldSearch(  
 searchValue: string,  
 searchQuery:...

REQ-3177: export \* from "./StaticValuePicker";

REQ-3178: export \* from "./FilterValuePicker";

REQ-3179: import { t } from "ttag";  
  
import type { ComboboxItem } from "metabase/ui";  
import type { FieldValuesSearchInfo } from "metabase-lib";  
import \* as Lib from "metabase-lib";  
import type { FieldValue,...

REQ-3180: export \* from "./MonthYearPicker";

REQ-3181: export \* from "./NumberFilterInput";

REQ-3182: export \* from "./QuarterYearPicker";

REQ-3183: export \* from "./RelativeDateShortcutPicker";

REQ-3184: import type { RelativeDatePickerValue } from "metabase/querying/filters/types";  
  
export type ShortcutGroup = {  
 label?: string;  
 columns: number;  
 shortcuts: Shortcut[];  
};  
  
export type Shortcut =...

REQ-3185: import { msgid, ngettext, t } from "ttag";  
  
import type { ShortcutGroup } from "./types";  
  
export function getShortcutGroups(): ShortcutGroup[] {  
 return [  
 {  
 columns: 2,  
 shortcuts:...

REQ-3186: export \* from "./TemporalUnitPicker";

REQ-3187: export \* from "./TimeseriesBucketPicker";

REQ-3188: export \* from "./TimeseriesFilterPicker";

REQ-3189: export \* from "./TimeseriesChrome";

REQ-3190: import \* as Lib from "metabase-lib";  
  
export function findFilterColumn(  
 query: Lib.Query,  
 stageIndex: number,  
 breakoutColumn: Lib.ColumnMetadata,  
): Lib.ColumnMetadata | undefined {  
 const...

REQ-3191: export const SPECIFIC\_DATE\_PICKER\_OPERATORS = [  
 "=" as const,  
 "<" as const,  
 ">" as const,  
 "between" as const,  
];  
  
export const EXCLUDE\_DATE\_PICKER\_OPERATORS = [  
 "!=" as const,  
 "is-null"...

REQ-3192: export \* from "./use-boolean-filter";

REQ-3193: import { useState } from "react";  
  
import type \* as Lib from "metabase-lib";  
  
import { getFilterClause, getFilterValue } from "./utils";  
  
type UseBooleanFilterProps = {  
 query: Lib.Query;  
 ...

REQ-3194: import { act, renderHook } from "@testing-library/react";  
  
import { createMockMetadata } from "\_\_support\_\_/metadata";  
import type { BooleanFilterValue } from "metabase/querying/filters/types";  
import...

REQ-3195: import type { BooleanFilterValue } from "metabase/querying/filters/types";  
import \* as Lib from "metabase-lib";  
  
export function getFilterValue(  
 query: Lib.Query,  
 stageIndex: number,  
 ...

REQ-3196: import type \* as Lib from "metabase-lib";  
  
import type { OperatorOption } from "./types";  
  
export const OPERATOR\_OPTIONS: Record<  
 Lib.CoordinateFilterOperator,  
 OperatorOption  
> = {  
 "=": {  
 ...

REQ-3197: export \* from "./use-coordinate-filter";  
export \* from "./types";

REQ-3198: import type { FilterOperatorOption } from "metabase/querying/filters/types";  
import type \* as Lib from "metabase-lib";  
  
type CoordinatePickerOperator =  
 | "="  
 | "!="  
 | ">"  
 | "<"  
 | "between"  
...

REQ-3199: import { useMemo, useState } from "react";  
  
import \* as Lib from "metabase-lib";  
  
import type { NumberOrEmptyValue } from "./types";  
import {  
 canPickColumns,  
 getAvailableColumns,  
 ...

REQ-3200: import { act, renderHook } from "@testing-library/react";  
  
import { checkNotNull } from "metabase/lib/types";  
import \* as Lib from "metabase-lib";  
import { columnFinder, createQuery } from...

REQ-3201: import { isNotNull } from "metabase/lib/types";  
import {  
 getAvailableOperatorOptions,  
 getDefaultAvailableOperator,  
} from "metabase/querying/filters/utils/operators";  
import \* as Lib from...

REQ-3202: export \* from "./use-date-filter";

REQ-3203: import { useMemo } from "react";  
  
import type { DatePickerValue } from "metabase/querying/filters/types";  
import {  
 getDateFilterClause,  
 getDatePickerOperators,  
 getDatePickerUnits,  
 ...

REQ-3204: import { renderHook } from "@testing-library/react";  
  
import type { DatePickerValue } from "metabase/querying/filters/types";  
import \* as Lib from "metabase-lib";  
import {  
 columnFinder,  
 ...

REQ-3205: import type \* as Lib from "metabase-lib";  
  
import type { OperatorOption } from "./types";  
  
export const OPERATOR\_OPTIONS: Record<  
 Lib.DefaultFilterOperator,  
 OperatorOption  
> = {  
 "is-null": {  
 ...

REQ-3206: export \* from "./use-default-filter";  
export \* from "./types";

REQ-3207: import type { FilterOperatorOption } from "metabase/querying/filters/types";  
import type \* as Lib from "metabase-lib";  
  
export type OperatorOption = FilterOperatorOption<Lib.DefaultFilterOperator>;

REQ-3208: import { useMemo, useState } from "react";  
  
import \* as Lib from "metabase-lib";  
  
import {  
 getAvailableOptions,  
 getDefaultOperator,  
 getFilterClause,  
} from "./utils";  
  
interface...

REQ-3209: import { act, renderHook } from "@testing-library/react";  
  
import { createMockMetadata } from "\_\_support\_\_/metadata";  
import { checkNotNull } from "metabase/lib/types";  
import \* as Lib from...

REQ-3210: import {  
 getAvailableOperatorOptions,  
 getDefaultAvailableOperator,  
} from "metabase/querying/filters/utils/operators";  
import \* as Lib from "metabase-lib";  
  
import { OPERATOR\_OPTIONS } from...

REQ-3211: import type \* as Lib from "metabase-lib";  
  
import type { OperatorOption } from "./types";  
  
export const OPERATOR\_OPTIONS: Record<  
 Lib.NumberFilterOperator,  
 OperatorOption  
> = {  
 "=": {  
 ...

REQ-3212: export \* from "./use-number-filter";  
export \* from "./types";

REQ-3213: import type { FilterOperatorOption } from "metabase/querying/filters/types";  
import type \* as Lib from "metabase-lib";  
  
export interface OperatorOption  
 extends...

REQ-3214: import { useMemo, useState } from "react";  
  
import \* as Lib from "metabase-lib";  
  
import type { NumberOrEmptyValue } from "./types";  
import {  
 getAvailableOptions,  
 getDefaultOperator,  
 ...

REQ-3215: import { act, renderHook } from "@testing-library/react";  
  
import { createMockMetadata } from "\_\_support\_\_/metadata";  
import { checkNotNull } from "metabase/lib/types";  
import \* as Lib from...

REQ-3216: import { isNotNull } from "metabase/lib/types";  
import {  
 getAvailableOperatorOptions,  
 getDefaultAvailableOperator,  
} from "metabase/querying/filters/utils/operators";  
import \* as Lib from...

REQ-3217: import type \* as Lib from "metabase-lib";  
  
import type { OperatorOption } from "./types";  
  
export const OPERATOR\_OPTIONS: Record<  
 Lib.StringFilterOperator,  
 OperatorOption  
> = {  
 "=": {  
 ...

REQ-3218: export \* from "./use-string-filter";  
export \* from "./types";

REQ-3219: import type { FilterOperatorOption } from "metabase/querying/filters/types";  
import type \* as Lib from "metabase-lib";  
  
export type OperatorType = "exact" | "partial" | "empty";  
  
export interface...

REQ-3220: import { useMemo, useState } from "react";  
  
import \* as Lib from "metabase-lib";  
  
import {  
 getAvailableOptions,  
 getDefaultOperator,  
 getDefaultValues,  
 getFilterClause,  
 getOptionByOperator,  
 ...

REQ-3221: import { act, renderHook } from "@testing-library/react";  
  
import { createMockMetadata } from "\_\_support\_\_/metadata";  
import { checkNotNull } from "metabase/lib/types";  
import \* as Lib from...

REQ-3222: import {  
 getAvailableOperatorOptions,  
 getDefaultAvailableOperator,  
} from "metabase/querying/filters/utils/operators";  
import \* as Lib from "metabase-lib";  
  
import { OPERATOR\_OPTIONS } from...

REQ-3223: import type \* as Lib from "metabase-lib";  
  
import type { OperatorOption } from "./types";  
  
export const OPERATOR\_OPTIONS: Record<Lib.TimeFilterOperator, OperatorOption> =  
 {  
 "<": {  
 ...

REQ-3224: export \* from "./use-time-filter";  
export \* from "./types";

REQ-3225: import type { FilterOperatorOption } from "metabase/querying/filters/types";  
import type \* as Lib from "metabase-lib";  
  
export interface OperatorOption  
 extends...

REQ-3226: import { useMemo, useState } from "react";  
  
import \* as Lib from "metabase-lib";  
  
import type { TimeValue } from "./types";  
import {  
 getAvailableOptions,  
 getDefaultOperator,  
 getDefaultValues,  
 ...

REQ-3227: import { act, renderHook } from "@testing-library/react";  
  
import { createMockMetadata } from "\_\_support\_\_/metadata";  
import { checkNotNull } from "metabase/lib/types";  
import \* as Lib from...

REQ-3228: import dayjs from "dayjs";  
  
import { isNotNull } from "metabase/lib/types";  
import {  
 getAvailableOperatorOptions,  
 getDefaultAvailableOperator,  
} from...

REQ-3229: import type { IconName } from "metabase/ui";  
import type \* as Lib from "metabase-lib";  
  
import type {  
 DATE\_PICKER\_EXTRACTION\_UNITS,  
 DATE\_PICKER\_SHORTCUTS,  
 DATE\_PICKER\_TRUNCATION\_UNITS,  
 ...

REQ-3230: import dayjs from "dayjs";  
import { match } from "ts-pattern";  
import { c, msgid, ngettext, t } from "ttag";  
  
import {  
 DATE\_PICKER\_EXTRACTION\_UNITS,  
 DATE\_PICKER\_OPERATORS,  
 ...

REQ-3231: import type { DateFilterValue } from "metabase/querying/filters/types";  
import \* as Lib from "metabase-lib";  
import { columnFinder, createQuery } from "metabase-lib/test-helpers";  
  
import {...

REQ-3232: import type \* as Lib from "metabase-lib";  
  
export function getGroupName(  
 groupInfo: Lib.ColumnGroupDisplayInfo,  
 stageIndex: number,  
) {  
 return groupInfo.isMainGroup && stageIndex > 1  
 ?...

REQ-3233: import type { FilterOperatorOption } from "metabase/querying/filters/types";  
import \* as Lib from "metabase-lib";  
  
export function getAvailableOperatorOptions<  
 T extends...

REQ-3234: export \* from "./MetricEditorSidebar";

REQ-3235: export \* from "./ResizableBoxHandle";

REQ-3236: export \* from "./MetricEditorBody";

REQ-3237: export \* from "./MetricEmptyState";

REQ-3238: export \* from "./MetricEditorFooter";

REQ-3239: export \* from "./MetricEditorHeader";

REQ-3240: export \* from "./MetricEditor";

REQ-3241: export type MetricModalType = "create" | "leave";

REQ-3242: export \* from "./AggregateStep";

REQ-3243: export \* from "./BreakoutStep";

REQ-3244: export \* from "./ClauseStep";

REQ-3245: export \* from "./DataStep";

REQ-3246: export \* from "./FieldPicker";

REQ-3247: export \* from "./FilterStep";

REQ-3248: export \* from "./Join";

REQ-3249: export \* from "./JoinComplete";

REQ-3250: export \* from "./JoinCondition";

REQ-3251: export \* from "./JoinColumnButton";

REQ-3252: export \* from "./JoinColumnDropdown";

REQ-3253: export \* from "./JoinConditionColumnPicker";

REQ-3254: export \* from "./JoinConditionDraft";

REQ-3255: import \* as Lib from "metabase-lib";  
  
export function getDefaultJoinConditionOperator(  
 query: Lib.Query,  
 stageIndex: number,  
): Lib.JoinConditionOperator {  
 const operators =...

REQ-3256: export \* from "./JoinConditionOperatorPicker";

REQ-3257: export \* from "./JoinConditionRemoveButton";

REQ-3258: export \* from "./JoinDraft";

REQ-3259: import \* as Lib from "metabase-lib";  
  
export function getDefaultJoinStrategy(  
 query: Lib.Query,  
 stageIndex: number,  
): Lib.JoinStrategy {  
 const strategies = Lib.availableJoinStrategies(query,...

REQ-3260: export \* from "./JoinStrategyPicker";

REQ-3261: export \* from "./JoinTableColumnDraftPicker";

REQ-3262: export \* from "./JoinTableColumnPicker";

REQ-3263: export \* from "./JoinTablePicker";

REQ-3264: export \* from "./JoinStep";

REQ-3265: import type { IconName } from "metabase/ui";  
import type \* as Lib from "metabase-lib";  
  
const JOIN\_ICONS: Record<string, IconName> = {  
 "left-join": "join\_left\_outer",  
 "right-join":...

REQ-3266: export { LimitStep } from "./LimitStep";

REQ-3267: export function isLimitValid(number: number) {  
 return !Number.isNaN(number) && Number.isInteger(number) && number > 0;  
}  
  
export function parseLimit(value: string) {  
 return parseInt(value, 10);  
}

REQ-3268: export \* from "./Notebook";  
export \* from "./use-run-visualization";  
export \* from "./VisualizationButton";

REQ-3269: import { cleanQuestion } from "metabase/query\_builder/utils/question";  
import type Question from "metabase-lib/v1/Question";  
  
type UseVisualizationProps = {  
 question?: Question;  
 isDirty:...

REQ-3270: export const CONTAINER\_PADDING = "10px";

REQ-3271: export \* from "./NotebookCell";

REQ-3272: export { EmbeddingDataPicker } from "./EmbeddingDataPicker";

REQ-3273: export \* from "./NotebookDataPicker";

REQ-3274: import \* as Urls from "metabase/lib/urls";  
import \* as Lib from "metabase-lib";  
  
type Props = {  
 query: Lib.Query;  
 table?: Lib.TableMetadata | Lib.CardMetadata;  
 stageIndex: number;  
};  
  
export...

REQ-3275: export { NotebookNativePreview } from "./NotebookNativePreview";

REQ-3276: import { formatNativeQuery } from "metabase/lib/engine";  
import type Question from "metabase-lib/v1/Question";  
import type { NativeDatasetResponse } from "metabase-types/api";  
  
export function...

REQ-3277: export \* from "./NotebookActionButton";

REQ-3278: export \* from "./NotebookStepHeader";

REQ-3279: export \* from "./NotebookStepPreview";

REQ-3280: export \* from "./NotebookStep";

REQ-3281: import type { ComponentType } from "react";  
import { t } from "ttag";  
  
import { color } from "metabase/lib/colors";  
import type { IconName } from "metabase/ui";  
  
import type {  
 ...

REQ-3282: export { NotebookStepList } from "./NotebookStepList";

REQ-3283: export \* from "./SortStep";

REQ-3284: export \* from "./SummarizeStepHeader";

REQ-3285: export \* from "./SummarizeStep";

REQ-3286: istanbul ignore file

REQ-3287: import type \* as Lib from "metabase-lib";  
import type Question from "metabase-lib/v1/Question";  
  
export type NotebookStepType =  
 | "data"  
 | "join"  
 | "expression"  
 | "filter"  
 | "summarize"  
 |...

REQ-3288: import \_ from "underscore";  
  
import type { Query } from "metabase-lib";  
import \* as Lib from "metabase-lib";  
import type Question from "metabase-lib/v1/Question";  
import type Metadata from...

REQ-3289: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import { checkNotNull } from "metabase/lib/types";  
import \* as Lib from "metabase-lib";  
import { createQueryWithClauses } from...

REQ-3290: export \* from "./BooleanWidget";

REQ-3291: export \* from "./DateAllOptionsWidget";

REQ-3292: export \* from "./DateMonthYearWidget";

REQ-3293: export \* from "./DateQuarterYearWidget";

REQ-3294: export \* from "./DateRangeWidget";

REQ-3295: export \* from "./DateRelativeWidget";

REQ-3296: export \* from "./DateSingleWidget";

REQ-3297: export \* from "./WidgetFooter";

REQ-3298: export const MIN\_WIDTH = 300;

REQ-3299: import dayjs from "dayjs";  
  
import { type NumberValue, parseNumber } from "metabase/lib/number";  
import type { DateFilterValue } from "metabase/querying/filters/types";  
import {...

REQ-3300: import type { DateFilterValue } from "metabase/querying/filters/types";  
import type { ParameterValueOrArray } from "metabase-types/api";  
  
import {  
 deserializeBooleanParameterValue,  
 ...

REQ-3301: import { P, match } from "ts-pattern";  
  
import { isNotNull } from "metabase/lib/types";  
import { getDateFilterClause } from "metabase/querying/filters/utils/dates";  
import \* as Lib from...

REQ-3302: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import { isNotNull } from "metabase/lib/types";  
import \* as Lib from "metabase-lib";  
import {  
 columnFinder,  
 createQuery,  
 ...

REQ-3303: export \* from "./ClauseStep";

REQ-3304: export \* from "./DataStep";

REQ-3305: export \* from "./FilterStep";

REQ-3306: export \* from "./PreviewStep";

REQ-3307: export \* from "./SegmentEditor";

REQ-3308: import \* as Lib from "metabase-lib";  
import {  
 getColumnKey,  
 getColumnNameFromKey,  
} from "metabase-lib/v1/queries/utils/column-key";  
import {  
 isColumnNameCollapsedRowsSetting,  
 ...

REQ-3309: import Questions from "metabase/entities/questions";  
import Tables from "metabase/entities/tables";  
import type { Card, TableId, UnsavedCard } from "metabase-types/api";  
import { isSavedCard } from...

REQ-3310: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3311: export const QUESTION\_NAME\_MAX\_LENGTH = 254;

REQ-3312: Reducers shared between "main" and "public" apps

REQ-3313: Reducers needed for main application

REQ-3314: Reducers needed for public questions and dashboards

REQ-3315: import {  
 type PayloadAction,  
 createAction,  
 createSlice,  
} from "@reduxjs/toolkit";  
import { LOCATION\_CHANGE, push } from "react-router-redux";  
  
import {  
 isSmallScreen,  
 openInBlankWindow,  
 ...

REQ-3316: import { createReducer } from "@reduxjs/toolkit";  
  
import { login, loginGoogle, pauseRedirect } from "metabase/auth/actions";  
  
const initialState = {  
 loginPending: false,  
 redirect:...

REQ-3317: import { trackSchemaEvent } from "metabase/lib/analytics";  
  
import type { ResourceAccessedVia, ResourceType } from "./downloads";  
  
const SCHEMA = "downloads";  
  
export const trackDownloadResults = ({  
...

REQ-3318: import { createSlice } from "@reduxjs/toolkit";  
import { t } from "ttag";  
import \_ from "underscore";  
  
import api, { GET, POST } from "metabase/lib/api";  
import { isWithinIframe, openSaveDialog }...

REQ-3319: import api from "metabase/lib/api";  
import Question from "metabase-lib/v1/Question";  
import { createMockCard } from "metabase-types/api/mocks";  
  
import { getChartFileName, getDatasetDownloadUrl }...

REQ-3320: import {  
 type PayloadAction,  
 asyncThunkCreator,  
 buildCreateSlice,  
} from "@reduxjs/toolkit";  
import { compose, pick } from "underscore";  
  
import { parseSearchOptions } from...

REQ-3321: import { type Dispatch, configureStore } from "@reduxjs/toolkit";  
  
import { reducer as embeddingDataPickerReducer } from "../embedding-data-picker";  
  
import {  
 ...

REQ-3322: export \* from "./embed";

REQ-3323: import type { PayloadAction } from "@reduxjs/toolkit";  
import { createSlice } from "@reduxjs/toolkit";  
import \_ from "underscore";  
  
import type {  
 EmbeddingDataPickerState,  
 EmbeddingEntityType,  
}...

REQ-3324: import \* as entitiesMap from "metabase/entities";  
import { combineEntities } from "metabase/lib/entities";  
  
const entitiesArray = Object.values(entitiesMap);  
  
export const { entities, reducer,...

REQ-3325: import { getIn } from "icepick";  
import \_ from "underscore";  
  
import { cardApi, dashboardApi, datasetApi } from "metabase/api";  
import Databases from "metabase/entities/databases";  
import Fields from...

REQ-3326: import Fields from "metabase/entities/fields";  
  
import { fetchField } from "./metadata";  
  
describe("deprecated metadata actions", () => {  
 let dispatch;  
  
 beforeEach(() => {  
 ...

REQ-3327: import { assoc, getIn, updateIn } from "icepick";  
import { createAction, handleActions } from "redux-actions";  
  
export const setRequestLoading = createAction(  
 ...

REQ-3328: import { assocIn } from "icepick";  
import \_ from "underscore";  
  
import { FETCH\_REVISIONS } from "./metadata";  
  
// NOTE: actions are still in metabase/redux/metadata  
  
export default (state = {},...

REQ-3329: import { createAction, createReducer } from "@reduxjs/toolkit";  
  
import { sessionApi } from "metabase/api";  
import { createAsyncThunk } from "metabase/lib/redux";  
import { SettingsApi } from...

REQ-3330: import { createAction, handleActions } from "metabase/lib/redux";  
  
export const SET\_OPEN\_MODAL = "metabase/ui/SET\_OPEN\_MODAL";  
export const CLOSE\_MODAL = "metabase/ui/CLOSE\_MODAL";  
  
export const...

REQ-3331: import { createSelector } from "@reduxjs/toolkit";  
import { createRef } from "react";  
import type { Action } from "redux-actions";  
import \_ from "underscore";  
  
import { createAction,...

REQ-3332: import { configureStore } from "@reduxjs/toolkit";  
  
import { act } from "\_\_support\_\_/ui";  
import type { Dispatch } from "metabase-types/store";  
  
import {  
 addUndo,  
 dismissAllUndo,  
 dismissUndo,  
 ...

REQ-3333: import { assocIn, dissocIn, updateIn } from "icepick";  
import { t } from "ttag";  
  
import { cardApi } from "metabase/api";  
import Collections from "metabase/entities/collections";  
import {...

REQ-3334: import fetchMock from "fetch-mock";  
  
import { getStore } from "\_\_support\_\_/entities-store";  
import { Api } from "metabase/api";  
import { mainReducers } from "metabase/reducers-main";  
import {...

REQ-3335: import { createAction, createReducer } from "@reduxjs/toolkit";  
  
import Dashboards from "metabase/entities/dashboards";  
import { createAsyncThunk } from "metabase/lib/redux";  
import {...

REQ-3336: import { assoc } from "icepick";  
  
import { createAction, handleActions } from "metabase/lib/redux";  
  
import { filterUntouchedFields, isEmptyObject } from "./utils.js";  
  
export const SET\_ERROR =...

REQ-3337: export { SegmentList } from "./SegmentList";

REQ-3338: import { screen } from "\_\_support\_\_/ui";  
import { createMockUser } from "metabase-types/api/mocks";  
  
import { setup } from "./setup";  
  
describe("SegmentList (OSS)", () => {  
 describe("Admins", () =>...

REQ-3339: import { screen } from "\_\_support\_\_/ui";  
import { createMockUser } from "metabase-types/api/mocks";  
  
import type { SetupOpts } from "./setup";  
import { setup as baseSetup } from "./setup";  
  
function...

REQ-3340: import { screen } from "\_\_support\_\_/ui";  
import { createMockUser } from "metabase-types/api/mocks";  
  
import type { SetupOpts } from "./setup";  
import { setup as baseSetup } from "./setup";  
  
function...

REQ-3341: import { createSelector } from "@reduxjs/toolkit";  
import { getIn } from "icepick";  
  
import Dashboards from "metabase/entities/dashboards";  
import { resourceListToMap } from...

REQ-3342: import dayjs from "dayjs";  
import { assoc } from "icepick";  
import { t } from "ttag";  
  
import { humanize, titleize } from "metabase/lib/formatting";  
import \* as Urls from "metabase/lib/urls";  
import...

REQ-3343: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import { separateTablesBySchema } from "metabase/reference/databases/TableList";  
import { databaseToForeignKeys, getQuestion } from...

REQ-3344: normalizr schema for use in actions/reducers

REQ-3345: import { trackSchemaEvent } from "metabase/lib/analytics";  
import type { SearchRequest, SearchResponse } from "metabase-types/api";  
  
type SearchRequestFilter = Pick<  
 SearchRequest,  
 |...

REQ-3346: export { DropdownSidebarFilter } from "./DropdownSidebarFilter";

REQ-3347: export \* from "./InfoText";

REQ-3348: export \* from "./SearchFilterDateDisplay";

REQ-3349: export \* from "./SearchFilterDatePicker";

REQ-3350: export { SearchFilterPopoverWrapper } from "./SearchFilterPopoverWrapper";

REQ-3351: export const DEFAULT\_ICON\_SIZE = 16;  
export const LARGE\_ICON\_SIZE = 20;

REQ-3352: export { Context } from "./Context";  
export { ItemIcon } from "./ItemIcon";  
export \* from "./constants";

REQ-3353: import type { WrappedResult } from "metabase/search/types";  
  
import type { IconComponentProps } from "./ItemIcon";  
  
export const isWrappedResult = (  
 item: IconComponentProps["item"],  
): item is...

REQ-3354: export { SearchResult } from "./SearchResult";  
export \* from "./components";  
export \* from "./SearchResult.styled";

REQ-3355: import type { WrappedResult } from "metabase/search/types";  
import { createMockSearchResult } from "metabase-types/api/mocks";  
  
export const createWrappedSearchResult = (  
 options:...

REQ-3356: export \* from "./SearchResultLink";

REQ-3357: export { SearchSidebar } from "./SearchSidebar";

REQ-3358: export { SearchUserPicker } from "./SearchUserPicker";  
export \* from "./SearchUserPicker.styled";

REQ-3359: export { ToggleSidebarFilter } from "./ToggleSidebarFilter";  
export type { ToggleSidebarFilterProps } from "./ToggleSidebarFilter";

REQ-3360: export { UserListElement } from "./UserListElement";

REQ-3361: export { UserNameDisplay } from "./UserNameDisplay";  
export type { UserNameDisplayProps } from "./UserNameDisplay";

REQ-3362: export \* from "./CreatedByFilter";

REQ-3363: export { NativeQueryFilter } from "./NativeQueryFilter";

REQ-3364: export \* from "./TypeFilter";

REQ-3365: import type { EnabledSearchModel } from "metabase-types/api";  
  
export const SearchFilterKeys = {  
 Type: "type",  
 Verified: "verified",  
 CreatedBy: "created\_by",  
 CreatedAt: "created\_at",  
 ...

REQ-3366: export const PAGE\_SIZE = 50;

REQ-3367: import type { Location } from "history";  
import type { ComponentType } from "react";  
  
import type { SearchFilterKeys } from "metabase/search/constants";  
import type { IconName } from...

REQ-3368: import { enabledSearchTypes } from "metabase/search/constants";  
import type { EnabledSearchModel } from "metabase-types/api";  
  
export function isEnabledSearchModelType(  
 value: unknown,  
): value is...

REQ-3369: import {  
 filterEnabledSearchTypes,  
 isEnabledSearchModelType,  
} from "metabase/search/utils";  
  
const TEST\_VALID\_VALUES = [  
 "collection",  
 "dashboard",  
 "card",  
 "database",  
 "table",  
 ...

REQ-3370: export {  
 isEnabledSearchModelType,  
 filterEnabledSearchTypes,  
} from "./enabled-search-type";

REQ-3371: export \* from "./search-location";  
export \* from "./user-search-params";  
export \* from "./enabled-search-type";

REQ-3372: export \* from "./search-location";

REQ-3373: import \_ from "underscore";  
  
import { SearchFilterKeys } from "metabase/search/constants";  
import type {  
 SearchAwareLocation,  
 URLSearchFilterQueryParams,  
} from "metabase/search/types";  
  
export...

REQ-3374: import { SearchFilterKeys } from "metabase/search/constants";  
import type { SearchAwareLocation } from "metabase/search/types";  
import {  
 getFiltersFromLocation,  
 getSearchTextFromLocation,  
 ...

REQ-3375: export { parseUserIdArray, stringifyUserIdArray } from "./user-search-params";

REQ-3376: import { isNotNull } from "metabase/lib/types";  
import type { SearchQueryParamValue } from "metabase/search/types";  
import type { UserId } from "metabase-types/api";  
  
export const parseUserIdArray =...

REQ-3377: import {  
 parseUserId,  
 parseUserIdArray,  
 stringifyUserIdArray,  
} from "./user-search-params";  
  
describe("parseUserIdArray", () => {  
 it("should return a UserId array when value is a string", ()...

REQ-3378: import type { Selector } from "@reduxjs/toolkit";  
import { createSelector } from "@reduxjs/toolkit";  
import type { Location } from "history";  
  
import {  
 getDashboard,  
 getDashboardId,  
 ...

REQ-3379: import { getEngineNativeType } from "metabase/lib/engine";  
import type DatabaseEntity from "metabase-lib/v1/metadata/Database";  
import type { Database } from "metabase-types/api";  
  
export const...

REQ-3380: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import { checkNotNull } from "metabase/lib/types";  
import type { Database } from "metabase-types/api";  
import { createMockDatabase } from...

REQ-3381: import { createSelector } from "@reduxjs/toolkit";  
  
import { isEmbeddingSdk } from "metabase/embedding-sdk/config";  
import { isWithinIframe } from "metabase/lib/dom";  
import type {...

REQ-3382: import type { State } from "metabase-types/store";  
import type { EmbeddingEntityType } from "metabase-types/store/embedding-data-picker";  
  
export const getEntityTypes = (state: State):...

REQ-3383: import { createSelector } from "@reduxjs/toolkit";  
import { normalize } from "normalizr";  
  
import { FieldSchema } from "metabase/schema";  
import Question from "metabase-lib/v1/Question";  
import...

REQ-3384: import { createMockEntitiesState } from "\_\_support\_\_/store";  
import { checkNotNull } from "metabase/lib/types";  
import { getMetadata } from "metabase/selectors/metadata";  
import Metadata from...

REQ-3385: import type { State } from "metabase-types/store";  
  
export const getLocation = (state: State) =>  
 state.routing?.locationBeforeTransitions ?? { pathname: "" };

REQ-3386: import {  
 createMockTokenFeatures,  
 createMockTokenStatus,  
} from "metabase-types/api/mocks";  
import {  
 createMockSettingsState,  
 createMockState,  
} from "metabase-types/store/mocks";  
  
import {  
 ...

REQ-3387: import { createSelector } from "@reduxjs/toolkit";  
  
import { getPlan } from "metabase/common/utils/plan";  
import type { TokenStatus, Version } from "metabase-types/api";  
import type { State } from...

REQ-3388: import type { State } from "metabase-types/store";  
  
export const currentOpenModal = (state: State) => state.modal;

REQ-3389: import { createSelector } from "@reduxjs/toolkit";  
  
import { PLUGIN\_APPLICATION\_PERMISSIONS } from "metabase/plugins";  
import type { State } from "metabase-types/store";  
  
export const getUser =...

REQ-3390: import { createMockUser } from "metabase-types/api/mocks";  
import { createMockState } from "metabase-types/store/mocks";  
  
import { getUserIsAdmin } from "./user";  
  
describe("metabase/selectors/user",...

REQ-3391: import { createSelector } from "reselect";  
  
import { isEmbeddingSdk } from "metabase/embedding-sdk/config";  
  
import { getIsEmbeddingIframe } from "./embed";  
import { getSetting } from...

REQ-3392: import { PLUGIN\_SELECTORS } from "metabase/plugins";  
import type { State } from "metabase-types/store";  
  
export function getWhiteLabeledLoadingMessageFactory(state: State) {  
 return...

REQ-3393: import {  
 getApplicationName,  
 getCanWhitelabel,  
 getIsWhiteLabeling,  
 getShowMetabaseLinks,  
 getWhiteLabeledLoadingMessageFactory,  
} from "..";  
  
import { setup } from...

REQ-3394: import {  
 getApplicationName,  
 getCanWhitelabel,  
 getIsWhiteLabeling,  
 getShowMetabaseLinks,  
 getWhiteLabeledLoadingMessageFactory,  
} from "..";  
  
import type { SetupOpts } from "./setup";  
import...

REQ-3395: import {  
 getApplicationName,  
 getCanWhitelabel,  
 getIsWhiteLabeling,  
 getShowMetabaseLinks,  
 getWhiteLabeledLoadingMessageFactory,  
} from "..";  
  
import type { SetupOpts } from "./setup";  
import...

REQ-3396: import \_ from "underscore";  
  
import api, { DELETE, GET, POST, PUT } from "metabase/lib/api";  
import { IS\_EMBED\_PREVIEW } from "metabase/lib/embed";  
import { PLUGIN\_API, PLUGIN\_CONTENT\_TRANSLATION }...

REQ-3397: import fetchMock from "fetch-mock";  
  
import { createMockEntitiesState } from "\_\_support\_\_/store";  
import { defer } from "metabase/lib/promise";  
import { getMetadata } from...

REQ-3398: import { createAction } from "@reduxjs/toolkit";  
import { t } from "ttag";  
  
import { createDatabase } from "metabase/admin/databases/database";  
import {  
 initializeSettings,  
 updateSetting,  
 ...

REQ-3399: import { trackSchemaEvent } from "metabase/lib/analytics";  
import type { SetupVersion } from "metabase-types/analytics/setup";  
import type { UsageReason } from "metabase-types/api";  
  
import type {...

REQ-3400: export \* from "./ActiveStep";

REQ-3401: export \* from "./CloudMigrationHelp";

REQ-3402: export \* from "./CompletedStep";

REQ-3403: export \* from "./DataUsageStep";

REQ-3404: export \* from "./DatabaseHelp";

REQ-3405: export \* from "./DatabaseStep";

REQ-3406: export { WelcomeStep } from "./steps/WelcomeStep";  
export { DataConnectionStep } from "./steps/DataConnectionStep";  
export { TableSelectionStep } from "./steps/TableSelectionStep";  
export {...

REQ-3407: import { t } from "ttag";  
  
import type { IconName } from "metabase/ui";  
  
export interface StepDefinition {  
 key: string;  
 title: string;  
 icon: IconName;  
 visibleInSidebar: boolean;  
}  
  
export...

REQ-3408: export \* from "./InactiveStep";

REQ-3409: export \* from "./InviteUserForm";

REQ-3410: export { LanguageSelector } from "./LanguageSelector";

REQ-3411: export \* from "./LanguageStep";

REQ-3412: export { LicenseTokenStep } from "./LicenseTokenStep";

REQ-3413: export \* from "./SettingsPage";

REQ-3414: export \* from "./Setup";

REQ-3415: export \* from "./SetupCardContainer.styled";

REQ-3416: export \* from "./SetupHelp";

REQ-3417: export \* from "./SetupSection";

REQ-3418: export { UsageQuestionStep } from "./UsageQuestionStep";

REQ-3419: export \* from "./UserForm";

REQ-3420: export \* from "./UserStep";

REQ-3421: export \* from "./WelcomePage";

REQ-3422: export type NumberedStepProps = { stepLabel: number };

REQ-3423: export const LOCALE\_TIMEOUT = 300;  
  
export const SUBSCRIBE\_URL =  
 "https://metabase.us10.list-manage.com/subscribe/post?u=869fec0e4689e8fd1db91e795&id=b9664113a8";  
export const SUBSCRIBE\_TOKEN =...

REQ-3424: export \* from "./actions";  
export \* from "./selectors";

REQ-3425: import { createReducer } from "@reduxjs/toolkit";  
  
import type { SetupState } from "metabase-types/store";  
  
import {  
 loadLocaleDefaults,  
 loadUserDefaults,  
 selectStep,  
 skipDatabase,  
 ...

REQ-3426: import { createSelector } from "@reduxjs/toolkit";  
  
import { isEEBuild } from "metabase/lib/utils";  
import { getSetting } from "metabase/selectors/settings";  
import type {  
 DatabaseData,  
 ...

REQ-3427: export type SetupStep =  
 | "welcome"  
 | "language"  
 | "user\_info"  
 | "usage\_question"  
 | "db\_connection"  
 | "license\_token"  
 | "data\_usage"  
 | "completed";

REQ-3428: import { useDispatch, useSelector } from "metabase/lib/redux";  
  
import { selectStep } from "./actions";  
import {  
 getIsSetupCompleted,  
 getIsStepActive,  
 getIsStepCompleted,  
} from...

REQ-3429: import { getIn } from "icepick";  
import \_ from "underscore";  
  
import MetabaseSettings from "metabase/lib/settings";  
import { UtilApi } from "metabase/services";  
import type { LocaleData } from...

REQ-3430: export \* from "./ComboChart";

REQ-3431: import twoBarsTwoAreasOneLineLinear from "./2-bars-2-areas-1-line-linear.json";  
import twoBarsTwoAreasOneLineLog from "./2-bars-2-areas-1-line-log.json";  
import twoBarsTwoAreasOneLinePower from...

REQ-3432: export \* from "./FunnelBarChart";

REQ-3433: import funnelBarCategorical from "./funnel-bar-categorical.json";  
import funnelBarOrderedRows from "./funnel-bar-ordered-rows.json";  
import funnelBarUnorderedRows from...

REQ-3434: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3435: import type { FunnelProps } from "metabase/static-viz/components/FunnelChart/FunnelChart";  
  
export const DEFAULT: FunnelProps = {  
 data: [  
 ["Visitors", 1000],  
 ["Started sign up", 300],  
 ...

REQ-3436: import type { NumberFormatOptions } from "metabase/static-viz/lib/numbers";  
import type { VisualizationSettings } from "metabase-types/api";  
  
export type Step = string | number;  
export type Measure =...

REQ-3437: import type { PolygonProps } from "@visx/shape/lib/shapes/Polygon";  
  
import { isNotNull } from "metabase/lib/types";  
import { CHAR\_SIZES\_FONT\_WEIGHT } from...

REQ-3438: import { merge } from "icepick";  
  
import type { FunnelDatum, FunnelSettings } from "../types";  
  
import {  
 calculateFunnelPolygonPoints,  
 calculateFunnelSteps,  
 reorderData,  
} from...

REQ-3439: export \* from "./funnel";  
export \* from "./margin";

REQ-3440: import { formatNumber } from "metabase/static-viz/lib/numbers";  
import {  
 measureTextHeight,  
 measureTextWidth,  
} from "metabase/static-viz/lib/text";  
  
import type { FunnelDatum, FunnelSettings }...

REQ-3441: import type { GaugeContainerProps } from "metabase/static-viz/components/Gauge/GaugeContainer";  
  
export const DEFAULT: Omit<GaugeContainerProps, "getColor"> = {  
 card: {  
 visualization\_settings:...

REQ-3442: import type { NumberFormatOptions } from "metabase/static-viz/lib/numbers";  
  
import {  
 GAUGE\_ARC\_ANGLE,  
 SEGMENT\_LABEL\_ANCHOR\_THRESHOLD\_ANGLE,  
} from "./constants";  
import {  
 ...

REQ-3443: export const LEGEND\_CIRCLE\_SIZE = 10;  
export const LEGEND\_CIRCLE\_MARGIN\_RIGHT = 4;  
export const LEGEND\_ITEM\_MARGIN\_RIGHT = 24;  
export const LEGEND\_ITEM\_MARGIN\_RIGHT\_GRID = 16;  
  
export const...

REQ-3444: export \* from "./Legend";

REQ-3445: import type { LegendItem } from "metabase/visualizations/echarts/cartesian/model/types";  
  
export type PositionedLegendItem = LegendItem & {  
 left: number;  
 top: number;  
 width?: number;  
};

REQ-3446: import { measureTextWidth } from "metabase/static-viz/lib/text";  
import type { LegendItem } from "metabase/visualizations/echarts/cartesian/model/types";  
import { truncateText } from...

REQ-3447: import allNegativeWithOther from "./all-negative-with-other.json";  
import allNegative from "./all-negative.json";  
import allSettings from "./all-settings.json";  
import allZeroMetric44847 from...

REQ-3448: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3449: export const ZERO = {  
 data: {  
 value: 0,  
 goal: 100000,  
 },  
 settings: {  
 format: {  
 number\_style: "currency",  
 currency: "USD",  
 currency\_style: "symbol",  
 decimals:...

REQ-3450: export type ProgressBarData = {  
 value: number;  
 goal: number;  
};

REQ-3451: import Color from "color";  
import { t } from "ttag";  
  
import { measureTextWidth } from "metabase/static-viz/lib/text";  
  
import type { ProgressBarData } from "./types";  
  
const createPalette = (color:...

REQ-3452: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3453: import type { StaticRowChartProps } from "metabase/static-viz/components/RowChart/RowChart";  
  
export const MULTIPLE\_SERIES: StaticRowChartProps = {  
 settings: {  
 "graph.dimensions":...

REQ-3454: import type { RowChartTheme } from "metabase/visualizations/shared/components/RowChart/types";  
import type { ColorGetter } from "metabase/visualizations/types";  
  
export const getStaticChartTheme = (  
...

REQ-3455: import type { CartesianChartColumns } from "metabase/visualizations/lib/graph/columns";  
import type { VisualizationSettings } from "metabase-types/api";  
  
// Uses inverse axis settings to have...

REQ-3456: export \* from "./SankeyChart";

REQ-3457: import defaultSettings from "./default-settings.json";  
  
export const data = { defaultSettings };

REQ-3458: export \* from "./ScalarChart";

REQ-3459: import twoScalars from "./two-scalars.json";  
  
export const data = {  
 twoScalars,  
};

REQ-3460: import autoYAxisExcludeZeroWithGoal from "./auto-y-axis-exclude-zero-with-goal.json";  
import bubbleSize from "./bubble-size.json";  
import customYAxisRangeWithColumnScaling from...

REQ-3461: export \* from "./SmartScalar";

REQ-3462: export \* from "./StaticVisualization";  
export \* from "./types";

REQ-3463: import type {  
 ComputedVisualizationSettings,  
 RenderingContext,  
} from "metabase/visualizations/types";  
import type { RawSeries } from "metabase-types/api";  
  
export interface StaticChartProps {  
 ...

REQ-3464: export \* from "./Text";

REQ-3465: import customColors from "./custom-colors.json";  
import customYAxisRangeWithColumnScaling from "./custom-y-axis-range-with-column-scaling.json";  
import dataLabelsColumnFormatting from...

REQ-3466: export const DIMENSION\_ACCESSORS = {  
 dimension: (row: any[]) => row[0],  
 metric: (row: any[]) => row[1],  
};

REQ-3467: export const CHAR\_SIZES\_FONT\_SIZE = 10;  
export const CHAR\_SIZES\_FONT\_WEIGHT = 400;  
  
// Generated with the mapping tool from here https://github.com/Evgenus/js-server-text-width  
export const...

REQ-3468: export \* from "./LegacyStaticChart";

REQ-3469: import "./mock-environment";  
import "fast-text-encoding";  
  
import { setPlatformAPI } from "echarts/core";  
import ReactDOMServer from "react-dom/server";  
  
// eslint-disable-next-line...

REQ-3470: import { colors } from "metabase/lib/colors/colors";  
import { color } from "metabase/lib/colors/palette";  
import type { ColorPalette } from "metabase/lib/colors/types";  
import type { ColorGetter }...

REQ-3471: import type { NumberLike, StringLike } from "@visx/scale";  
  
import { formatValue } from "metabase/lib/formatting";  
import { getFormattingOptionsWithoutScaling } from...

REQ-3472: import { formatNumber as appFormatNumber } from "metabase/lib/formatting/numbers";  
  
export type NumberFormatOptions = {  
 number\_style?: "currency" | "decimal" | "scientific" | "percentage";  
 ...

REQ-3473: import { formatNumber, formatPercent } from "./numbers";  
  
describe("formatNumber", () => {  
 it("should format a number with default options", () => {  
 const number = 1500;  
  
 const text =...

REQ-3474: import type { ColorPalette } from "metabase/lib/colors/types";  
import { DEFAULT\_VISUALIZATION\_THEME } from "metabase/visualizations/shared/utils/theme";  
import type { RenderingContext } from...

REQ-3475: import type { Element, ElementContent } from "hast";  
import { fromHtml } from "hast-util-from-html";  
import { toHtml } from "hast-util-to-html";  
  
// FIXME: instead of Regex parse svg, update,...

REQ-3476: import type { Element, ElementContent } from "hast";  
import { fromHtml } from "hast-util-from-html";  
  
import { patchDominantBaseline } from "./svg";  
  
const OUTER\_TEXT = "outer-text";  
const G\_ELEM =...

REQ-3477: import { init } from "server-text-width";  
  
import {  
 CHAR\_SIZES,  
 CHAR\_SIZES\_FONT\_SIZE,  
 CHAR\_SIZES\_FONT\_WEIGHT,  
} from "../constants/char-sizes";  
  
const FONT\_WEIGHT\_WIDTH\_FACTOR = 0.039;  
  
export...

REQ-3478: import { measureTextWidth } from "./text";  
  
const fontSize = 11;  
  
describe("measureTextWidth", () => {  
 it("should measure text", () => {  
 expect(Math.round(measureTextWidth("abc",...

REQ-3479: eslint-disable no-prototype-builtins

REQ-3480: import {  
 registerVisualization,  
 setDefaultVisualization,  
} from "metabase/visualizations";  
import { AreaChart } from "metabase/visualizations/visualizations/AreaChart";  
import { BarChart } from...

REQ-3481: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3482: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3483: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3484: export \* from "./DownloadsStatus";

REQ-3485: export \* from "./DownloadsStatusLarge";

REQ-3486: export \* from "./DownloadsStatusSmall";

REQ-3487: export \* from "./FileUploadStatus";

REQ-3488: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3489: export \* from "./PublicStatusListing";

REQ-3490: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3491: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3492: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3493: import type { Download } from "metabase-types/store";  
  
export const isCompleted = (download: Download) =>  
 download.status === "complete";  
export const isErrored = (download: Download) =>...

REQ-3494: import { isReducedMotionPreferred } from "metabase/lib/dom";  
import type { IconName } from "metabase/ui";  
import type { LongTaskStatus } from "metabase-types/api";  
  
export const getIconName =...

REQ-3495: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3496: import { useBeforeUnload } from "react-use";  
import { t } from "ttag";  
  
import { useSelector } from "metabase/lib/redux";  
import { hasActiveDownloads } from "metabase/redux/downloads";  
  
export const...

REQ-3497: import { useLayoutEffect, useState } from "react";  
  
export const HIDE\_DELAY = 6000;  
  
const useStatusVisibility = (isActive: boolean) => {  
 const [isVisible, setIsVisible] = useState(isActive);  
  
 ...

REQ-3498: import { combineReducers, configureStore } from "@reduxjs/toolkit";  
import { routerMiddleware, routerReducer as routing } from "react-router-redux";  
import promise from "redux-promise";  
  
import { Api...

REQ-3499: export \* from "./EmotionCacheProvider";

REQ-3500: export { GlobalStyles } from "./GlobalStyles";

REQ-3501: import { createSelector } from "@reduxjs/toolkit";  
import \_ from "underscore";  
  
import { getEmbedOptions } from "metabase/selectors/embed";  
import { getSettings } from...

REQ-3502: export const SPACE\_LEVELS = ["4px", "8px", "16px", "32px", "64px", "128px"];

REQ-3503: eslint-disable-next-line no-restricted-imports

REQ-3504: import type { MantineTheme } from "metabase/ui";  
  
import { getThemeSpecificCssVariables } from "./css-variables";  
  
describe("getThemeSpecificCssVariables", () => {  
 it("returns the correct CSS...

REQ-3505: export \* from "./media-queries";  
export \* from "./space";  
export \* from "./typography";

REQ-3506: export const breakpointMinExtraSmall = "@media screen and (min-width: 23em)";  
export const breakpointMinSmall = "@media screen and (min-width: 40em)";  
export const breakpointMinMedium = "@media...

REQ-3507: import { SPACE\_LEVELS as levels } from "./constants";  
  
/\*\*  
 \* Returns a pixel amount: 4px, 8px, 16px, on to 128px  
 \* @param {number} level must be an integer between 0 and 5  
 \* @returns {string}  
...

REQ-3508: import { space } from "./space";  
  
it("returns pixel amount for acceptable levels", () => {  
 expect(space(0)).toBe("4px");  
 expect(space(1)).toBe("8px");  
 expect(space(2)).toBe("16px");  
 ...

REQ-3509: export const monospaceFontFamily = `"Monaco", "Menlo", "Ubuntu Mono", "Consolas", "Source Code Pro", "source-code-pro", monospace`;

REQ-3510: import { push } from "react-router-redux";  
  
import Timelines from "metabase/entities/timelines";  
import \* as Urls from "metabase/lib/urls";  
import type { Timeline } from "metabase-types/api";  
import...

REQ-3511: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3512: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3513: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3514: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3515: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3516: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3517: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3518: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3519: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3520: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3521: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3522: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3523: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3524: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3525: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3526: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3527: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3528: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3529: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3530: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3531: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3532: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3533: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3534: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3535: export interface MenuItem {  
 title: string;  
 link?: string;  
 action?: () => void;  
}  
  
export interface ModalParams {  
 slug: string;  
 timelineId?: string;  
 timelineEventId?: string;  
}

REQ-3536: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3537: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3538: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3539: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3540: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3541: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3542: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3543: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3544: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3545: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3546: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3547: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3548: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3549: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3550: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3551: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3552: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3553: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3554: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3555: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3556: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3557: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3558: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3559: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3560: export { actionIconOverrides } from "./ActionIcon.config";  
export { ActionIcon } from "@mantine/core";  
export type { ActionIconProps } from "@mantine/core";

REQ-3561: import type { ButtonProps as MantineButtonProps } from "@mantine/core";  
import type { HTMLAttributes } from "react";  
  
export { Button } from "@mantine/core";  
export type { ButtonGroupProps } from...

REQ-3562: export { CopyButton, type CopyButtonProps } from "@mantine/core";

REQ-3563: export \* from "./PopoverBackButton";

REQ-3564: export { UnstyledButton } from "@mantine/core";  
export type { UnstyledButtonProps } from "@mantine/core";

REQ-3565: export \* from "./ActionIcon";  
export \* from "./Button";  
export \* from "./PopoverBackButton";  
export \* from "./UnstyledButton";  
export \* from "./CopyButton";

REQ-3566: import { Accordion } from "@mantine/core";  
  
import AccordionStyles from "./Accordion.module.css";  
  
export const accordionOverrides = {  
 Accordion: Accordion.extend({  
 classNames: {  
 control:...

REQ-3567: export { Accordion } from "@mantine/core";  
export type { AccordionProps } from "@mantine/core";  
export { accordionOverrides } from "./Accordion.config";

REQ-3568: export { Badge } from "@mantine/core";  
export \* from "./Badge.config";

REQ-3569: import { Card } from "@mantine/core";  
  
import CardStyles from "./Card.module.css";  
  
export const cardOverrides = {  
 Card: Card.extend({  
 classNames: {  
 section: CardStyles.section,  
 },  
 ...

REQ-3570: export { Card } from "@mantine/core";  
export type { CardProps } from "@mantine/core";  
export { cardOverrides } from "./Card.config";

REQ-3571: export { Indicator } from "@mantine/core";

REQ-3572: import { Kbd } from "@mantine/core";  
  
import S from "./Kbd.module.css";  
  
export const kbdOverrides = {  
 Kbd: Kbd.extend({  
 classNames: {  
 root: S.root,  
 },  
 }),  
};

REQ-3573: export { Kbd } from "@mantine/core";  
  
export \* from "./Kbd.config";

REQ-3574: export \* from "./Accordion";  
export \* from "./Badge";  
export \* from "./Card";  
export \* from "./Image";  
export \* from "./Indicator";  
export \* from "./Kbd";

REQ-3575: export { Alert } from "@mantine/core";  
export type { AlertProps } from "@mantine/core";  
export { alertOverrides } from "./Alert.config";

REQ-3576: export { Loader } from "./Loader";  
export type { LoaderProps } from "@mantine/core";

REQ-3577: export { Progress } from "@mantine/core";  
export \* from "./Progress.config";

REQ-3578: import { Skeleton } from "@mantine/core";  
  
import S from "./Skeleton.module.css";  
  
export const skeletonOverrides = {  
 Skeleton: Skeleton.extend({  
 classNames: {  
 root: S.Skeleton,  
 },  
 ...

REQ-3579: export { Repeat } from "./Repeat";  
export { Skeleton } from "./Skeleton";  
export \* from "./Skeleton.config";

REQ-3580: export \* from "./Loader";  
export \* from "./Alert";  
export \* from "./Progress";  
export \* from "./Skeleton";

REQ-3581: import ten\_thousand\_component from "./10k.svg?component";  
import ten\_thousand\_source from "./10k.svg?source";  
import one\_million\_component from "./1m.svg?component";  
import one\_million\_source from...

REQ-3582: export \* from "./Icon";  
export \* from "./icons";

REQ-3583: export \* from "./Icon";

REQ-3584: export \* from "./buttons";  
export \* from "./data-display";  
export \* from "./feedback";  
export \* from "./icons";  
export \* from "./inputs";  
export \* from "./layout";  
export \* from...

REQ-3585: export { Autocomplete } from "@mantine/core";  
export type { AutocompleteProps } from "@mantine/core";  
export { autocompleteOverrides } from "./Autocomplete.config";

REQ-3586: export type { DateValue, DatesRangeValue } from "@mantine/dates";  
export { calendarOverrides } from "./Calendar.config";

REQ-3587: import { Checkbox, getSize, rem } from "@mantine/core";  
  
import CheckboxStyles from "./Checkbox.module.css";  
import { CheckboxIcon } from "./CheckboxIcon";  
  
const SIZES: Record<string, string> = {  
 ...

REQ-3588: export { Checkbox } from "@mantine/core";  
export type { CheckboxProps, CheckboxGroupProps } from "@mantine/core";  
export { checkboxOverrides } from "./Checkbox.config";

REQ-3589: import { Chip, type MantineThemeOverride } from "@mantine/core";  
  
import S from "./Chip.module.css";  
  
export const chipOverrides: MantineThemeOverride["components"] = {  
 Chip: Chip.extend({  
 ...

REQ-3590: export { Chip } from "@mantine/core";  
export type { ChipProps, ChipGroupProps } from "@mantine/core";  
export { chipOverrides } from "./Chip.config";

REQ-3591: import {  
 Combobox,  
 ComboboxChevron,  
 type MantineThemeOverride,  
} from "@mantine/core";  
  
import S from "./Combobox.module.css";  
  
export const comboboxOverrides:...

REQ-3592: export { Combobox } from "@mantine/core";  
export type {  
 ComboboxGroupProps,  
 ComboboxProps,  
 ComboboxItem,  
} from "@mantine/core";  
export { comboboxOverrides } from "./Combobox.config";

REQ-3593: export { DateInput } from "@mantine/dates";  
export type { DateInputProps } from "@mantine/dates";  
export { dateInputOverrides } from "./DateInput.config";

REQ-3594: export { DatePicker } from "@mantine/dates";  
export type { DatePickerProps } from "@mantine/dates";  
export { datePickerOverrides } from "./DatePicker.config";

REQ-3595: export { FileInput } from "@mantine/core";  
export type { FileInputProps } from "@mantine/core";  
export { fileInputOverrides } from "./FileInput.config";

REQ-3596: export { Input } from "@mantine/core";  
export { inputOverrides } from "./Input.config";

REQ-3597: export { MonthPicker } from "@mantine/dates";  
export type { MonthPickerProps } from "@mantine/dates";  
export { monthPickerOverrides } from "./MonthPicker.config";

REQ-3598: export \* from "./MultiAutocompleteOption";

REQ-3599: export \* from "./MultiAutocompleteValue";

REQ-3600: export \* from "./MultiAutocomplete";  
export \* from "./MultiAutocompleteOption";  
export \* from "./MultiAutocompleteValue";

REQ-3601: export \* from "./use-multi-autocomplete";

REQ-3602: import {  
 type ComboboxData,  
 type ComboboxItem,  
 type ComboboxParsedItem,  
 getParsedComboboxData,  
 isOptionsGroup,  
 useCombobox,  
} from "@mantine/core";  
import { useWindowEvent } from...

REQ-3603: export { MultiSelect, type MultiSelectProps } from "@mantine/core";  
export { multiSelectOverrides } from "./MultiSelect.config";

REQ-3604: export { NumberInput } from "./NumberInput";  
export type { NumberInputProps } from "./NumberInput";

REQ-3605: import { type MantineThemeOverride, Pill } from "@mantine/core";  
  
import S from "./Pill.module.css";  
  
export const pillOverrides: MantineThemeOverride["components"] = {  
 Pill: Pill.extend({  
 ...

REQ-3606: export { Pill } from "@mantine/core";  
export { pillOverrides } from "./Pill.config";

REQ-3607: import {  
 type MantineThemeOverride,  
 PillsInput,  
 PillsInputField,  
} from "@mantine/core";  
  
import S from "./PillsInput.module.css";  
  
export const pillsInputOverrides:...

REQ-3608: export { pillsInputOverrides } from "./PillsInput.config";

REQ-3609: export \* from "./QuarterPicker";

REQ-3610: import { Radio, getSize, rem } from "@mantine/core";  
  
import RadioStyles from "./Radio.module.css";  
  
const SIZES: Record<string, string> = {  
 md: rem(20),  
};  
  
export const radioOverrides = {  
 ...

REQ-3611: import {  
 type MantineTheme,  
 type MantineThemeOverride,  
 SegmentedControl,  
 type SegmentedControlProps,  
 rem,  
} from "@mantine/core";  
  
import S from "./SegmentedControl.module.css";  
  
export...

REQ-3612: export \* from "./DefaultSelectItem";  
export \* from "./SelectItem";  
export \* from "./utils";

REQ-3613: import { type MantineSize, getSize, rem } from "@mantine/core";  
  
const FONT\_SIZES = {  
 xs: rem(12),  
 md: rem(14),  
};  
  
const LINE\_HEIGHTS = {  
 xs: rem(16),  
 md: rem(24),  
};  
  
export function...

REQ-3614: export { Select, type SelectOption, type SelectProps } from "./Select";  
export { selectOverrides } from "./Select.config";  
export \* from "./SelectItem";

REQ-3615: import { Switch, getSize, rem } from "@mantine/core";  
  
import SwitchStyles from "./Switch.module.css";  
  
const LABEL\_FONT\_SIZES: Record<string, string> = {  
 xs: rem(12),  
 sm: rem(14),  
 md:...

REQ-3616: export { Switch } from "@mantine/core";  
export type { SwitchProps, SwitchGroupProps } from "@mantine/core";  
export { switchOverrides } from "./Switch.config";

REQ-3617: export { TextInput } from "@mantine/core";  
export type { TextInputProps } from "@mantine/core";  
export { textInputOverrides } from "./TextInput.config";

REQ-3618: export { Textarea } from "@mantine/core";  
export type { TextareaProps } from "@mantine/core";  
export { textareaOverrides } from "./Textarea.config";

REQ-3619: export \* from "./TimeInput";  
export { timeInputOverrides } from "./TimeInput.config";

REQ-3620: export \* from "./Autocomplete";  
export \* from "./Calendar";  
export \* from "./Checkbox";  
export \* from "./Chip";  
export \* from "./Combobox";  
export \* from "./DateInput";  
export \* from...

REQ-3621: export { Center } from "@mantine/core";  
export type { CenterProps } from "@mantine/core";

REQ-3622: export { Collapse } from "@mantine/core";  
export type { CollapseProps } from "@mantine/core";

REQ-3623: export { Flex } from "@mantine/core";  
export type { FlexProps } from "@mantine/core";

REQ-3624: export { Grid } from "@mantine/core";  
export type { GridProps } from "@mantine/core";

REQ-3625: export { Group } from "@mantine/core";  
export type { GroupProps } from "@mantine/core";

REQ-3626: export { ScrollArea } from "@mantine/core";  
export type { ScrollAreaProps } from "@mantine/core";  
  
export { scrollAreaOverrides } from "./ScrollArea.config";

REQ-3627: export { SimpleGrid } from "@mantine/core";  
export type { SimpleGridProps } from "@mantine/core";

REQ-3628: export { Stack } from "@mantine/core";  
export type { StackProps } from "@mantine/core";

REQ-3629: export \* from "./Center";  
export \* from "./Collapse";  
export \* from "./Flex";  
export \* from "./Grid";  
export \* from "./Group";  
export \* from "./SimpleGrid";  
export \* from "./ScrollArea";  
export \*...

REQ-3630: import { NavLink } from "@mantine/core";  
  
import S from "./NavLink.module.css";  
  
export const navLinkOverrides = {  
 NavLink: NavLink.extend({  
 defaultProps: {  
 //@ts-expect-error - this does...

REQ-3631: export { NavLink, type NavLinkProps } from "@mantine/core";  
  
export { navLinkOverrides } from "./NavLink.config";

REQ-3632: export { Tabs } from "@mantine/core";  
export type {  
 TabsProps,  
 TabsTabProps,  
 TabsListProps,  
 TabsPanelProps,  
} from "@mantine/core";  
export { tabsOverrides } from "./Tabs.config";

REQ-3633: export \* from "./NavLink";  
export \* from "./Tabs";

REQ-3634: export { Drawer } from "@mantine/core";

REQ-3635: import { Menu } from "@mantine/core";  
  
import MenuStyles from "./Menu.module.css";  
  
export const menuOverrides = {  
 Menu: Menu.extend({  
 defaultProps: {  
 radius: "sm",  
 shadow: "md",  
 ...

REQ-3636: export \* from "./MenuDropdown";

REQ-3637: export \* from "./MenuItem";

REQ-3638: export type { MenuProps, MenuItemProps } from "@mantine/core";  
export { Menu } from "./Menu";  
export { menuOverrides } from "./Menu.config";

REQ-3639: import { type MantineThemeOverride, Overlay } from "@mantine/core";  
  
import OverlayStyles from "./Overlay.module.css";  
  
export const overlayOverrides: MantineThemeOverride["components"] = {  
 ...

REQ-3640: import { Popover } from "@mantine/core";  
  
import PopoverStyles from "./Popover.module.css";  
  
export const DEFAULT\_POPOVER\_Z\_INDEX = 300;  
  
export const popoverOverrides = {  
 Popover:...

REQ-3641: export { Tooltip } from "@mantine/core";  
export type { TooltipProps } from "@mantine/core";  
export { tooltipOverrides } from "./Tooltip.config";

REQ-3642: export \* from "./Drawer";  
export \* from "./HoverCard";  
export \* from "./Menu";  
export \* from "./Modal";  
export \* from "./Popover";  
export \* from "./Overlay";  
export \* from "./Tooltip";

REQ-3643: export \* from "./DatesProvider";

REQ-3644: import { createContext } from "react";  
  
interface ThemeContext {  
 withCssVariables?: boolean;  
 withGlobalClasses?: boolean;  
}  
  
export const ThemeProviderContext = createContext<ThemeContext>({});

REQ-3645: export \* from "./ThemeProvider";

REQ-3646: export \* from "./ThemeProvider";

REQ-3647: export { Anchor } from "@mantine/core";  
export type { AnchorProps } from "@mantine/core";  
export { anchorOverrides } from "./Anchor.config";

REQ-3648: import { Code } from "@mantine/core";  
  
import CodeStyles from "./Code.module.css";  
  
export const codeOverrides = {  
 Code: Code.extend({  
 classNames: {  
 root: CodeStyles.root,  
 },  
 }),  
};

REQ-3649: export { Code } from "@mantine/core";  
export type { CodeProps } from "@mantine/core";  
export { codeOverrides } from "./Code.config";

REQ-3650: export { List } from "@mantine/core";  
export type { ListProps } from "@mantine/core";  
export { listOverrides } from "./List.config";

REQ-3651: export { Text } from "@mantine/core";  
export type { TextProps } from "@mantine/core";  
export { textOverrides } from "./Text.config";

REQ-3652: export { Title } from "@mantine/core";  
export type { TitleProps } from "@mantine/core";  
export { titleOverrides } from "./Title.config";

REQ-3653: export \* from "./Text";  
export \* from "./Title";  
export \* from "./Anchor";  
export \* from "./List";  
export \* from "./Code";

REQ-3654: export { Box } from "@mantine/core";  
export type { BoxProps } from "@mantine/core";

REQ-3655: export { Divider } from "@mantine/core";  
export type { DividerProps } from "@mantine/core";  
export { dividerOverrides } from "./Divider.config";

REQ-3656: export { FocusTrap } from "@mantine/core";  
export type { FocusTrapProps } from "@mantine/core";

REQ-3657: export { Paper } from "@mantine/core";  
export type { PaperProps } from "@mantine/core";  
export { paperOverrides } from "./Paper.config";

REQ-3658: export \* from "./PreventPopoverExit";

REQ-3659: export { Space } from "@mantine/core";

REQ-3660: export { Transition } from "@mantine/core";  
export type { TransitionProps } from "@mantine/core";

REQ-3661: export \* from "./Box";  
export \* from "./DelayGroup";  
export \* from "./Divider";  
export \* from "./FocusTrap";  
export \* from "./PreventEagerPortal";  
export \* from "./Paper";  
export \* from...

REQ-3662: export { rem, useCombobox, useMantineTheme } from "@mantine/core";  
export type {  
 FloatingPosition,  
 MantineSize,  
 MantineStyleProps,  
 MantineTheme,  
 MantineThemeOther,  
 MantineThemeOverride,  
}...

REQ-3663: import { HighlightStyle } from "@codemirror/language";  
import type { Tag } from "@lezer/highlight";  
import { tags } from "@lezer/highlight";  
  
import S from "./highlight.module.css";  
  
const styledTags...

REQ-3664: export \* from "./highlight";

REQ-3665: import type { MantineThemeOverride } from "@mantine/core";  
import { rem } from "@mantine/core";  
  
import { DEFAULT\_METABASE\_COMPONENT\_THEME } from "metabase/embedding-sdk/theme";  
  
import Styles from...

REQ-3666: import type { MantineTheme } from "@mantine/core";  
  
import { color as legacyColor } from "metabase/lib/colors";  
type ColorShades = MantineTheme["colors"]["dark"];  
  
const ORIGINAL\_COLORS = [  
 ...

REQ-3667: import type { MetabasePluginsConfig } from "metabase/embedding-sdk/types/plugins";  
import { queryDrill } from "metabase/querying/drills/utils/query-drill";  
import type { DrillThruDisplayInfo } from...

REQ-3668: export { Mode } from "./Mode";

REQ-3669: export { ColumnFormattingAction } from "./ColumnFormattingAction";

REQ-3670: export { CombineColumnsAction } from "./CombineColumnsAction";

REQ-3671: export { ExtractColumnAction } from "./ExtractColumnAction";

REQ-3672: export { HideColumnAction } from "./HideColumnAction";

REQ-3673: import type { MetabasePluginsConfig } from "metabase/embedding-sdk/types/plugins";  
import type { QueryClickActionsMode } from "metabase/visualizations/types";  
import type Question from...

REQ-3674: import type { QueryClickActionsMode } from "../../types";  
  
export const ArchivedMode: QueryClickActionsMode = {  
 name: "archived",  
 hasDrills: false,  
 clickActions: [],  
};

REQ-3675: import type { QueryClickActionsMode } from "../../types";  
import { ColumnFormattingAction } from "../actions/ColumnFormattingAction";  
import { CombineColumnsAction } from...

REQ-3676: import type { QueryClickActionsMode } from "../../types";  
import { CombineColumnsAction } from "../actions/CombineColumnsAction";  
import { DashboardClickAction } from...

REQ-3677: import type { QueryClickActionsMode } from "../../types";  
import { DashboardClickAction } from "../actions/DashboardClickAction";  
  
export const PublicMode: QueryClickActionsMode = {  
 name:...

REQ-3678: export \* from "./ChartRenderingErrorBoundary";

REQ-3679: import { useMemo, useState } from "react";  
import { t } from "ttag";  
import \_ from "underscore";  
  
import type { Widget } from "../types";  
  
import type { BaseChartSettingsProps } from "./types";  
  
//...

REQ-3680: export \* from "./BaseChartSettings";

REQ-3681: import type { StackProps } from "metabase/ui";  
import type { ComputedVisualizationSettings } from "metabase/visualizations/types";  
import type Question from "metabase-lib/v1/Question";  
import type {...

REQ-3682: export \* from "./ChartSettingsFooter";

REQ-3683: export \* from "./ChartSettingsVisualization";

REQ-3684: import type { StackProps } from "metabase/ui";  
import type Question from "metabase-lib/v1/Question";  
import type {  
 Dashboard,  
 DashboardCard,  
 RawSeries,  
 VisualizationSettings,  
} from...

REQ-3685: export \* from "./DashboardChartSettings";

REQ-3686: import type { Dashboard, DashboardCard } from "metabase-types/api";  
  
import type { CommonChartSettingsProps, Widget } from "../types";  
  
export type DashboardChartSettingsProps = {  
 className?:...

REQ-3687: export \* from "./QuestionChartSettings";

REQ-3688: import type { BaseChartSettingsProps } from "../BaseChartSettings/types";  
import type { CommonChartSettingsProps, Widget } from "../types";  
  
export type QuestionChartSettingsProps = {  
 widgets?:...

REQ-3689: import { assocIn } from "icepick";  
import { useCallback, useMemo } from "react";  
  
import {  
 extractRemappings,  
 getVisualizationTransformed,  
} from "metabase/visualizations";  
import {...

REQ-3690: export { QuestionChartSettings } from "./QuestionChartSettings";  
export { DashboardChartSettings } from "./DashboardChartSettings";  
export { BaseChartSettings } from "./BaseChartSettings";  
export \*...

REQ-3691: import type Question from "metabase-lib/v1/Question";  
import type { Series, VisualizationSettings } from "metabase-types/api";  
  
// this type is not full, we need to extend it later  
export type Widget...

REQ-3692: export \* from "./EChartsTooltip";

REQ-3693: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3694: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3695: import { t } from "ttag";  
  
import type { TooltipRowModel } from "metabase/visualizations/types";  
  
export const getTotalValue = (  
 headerRows: TooltipRowModel[] = [],  
 bodyRows: TooltipRowModel[] =...

REQ-3696: export \* from "./TooltipRow";

REQ-3697: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3698: import { NULL\_DISPLAY\_VALUE } from "metabase/lib/constants";  
import { formatValue } from "metabase/lib/formatting";  
import type { OptionsType } from "metabase/lib/formatting/types";  
import type {  
 ...

REQ-3699: import { mockIsEmbeddingSdk } from "metabase/embedding-sdk/mocks/config-mock";  
import {  
 getLegendTitles,  
 getMapUrl,  
} from...

REQ-3700: export { ClickActionsView } from "./ClickActionsView";  
export { ConnectedClickActionsPopover } from "./ClickActionsPopover";

REQ-3701: import { t } from "ttag";  
import \_ from "underscore";  
  
import type {  
 ClickActionSection,  
 RegularClickAction,  
} from "metabase/visualizations/types";  
  
type Section = {  
 index?: number;  
};  
  
export...

REQ-3702: export \* from "./EChartsRenderer";  
export type { ResponsiveEChartsRendererProps } from "./ResponsiveEChartsRenderer";  
export { ResponsiveEChartsRenderer } from "./ResponsiveEChartsRenderer.styled";

REQ-3703: export { EmptyVizState } from "./EmptyVizState";

REQ-3704: import { t } from "ttag";  
  
import { getSubpathSafeUrl } from "metabase/lib/urls";  
import type { CardDisplayType } from "metabase-types/api";  
  
/\*\*  
 \* The "table" and the "object" (detail) charts can...

REQ-3705: import { render, screen } from "\_\_support\_\_/ui";  
import LegendVertical from "metabase/visualizations/components/LegendVertical";  
  
describe("LegendVertical", () => {  
 it("should render string titles...

REQ-3706: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3707: import type Question from "metabase-lib/v1/Question";  
import type ForeignKey from "metabase-lib/v1/metadata/ForeignKey";  
import type Metadata from "metabase-lib/v1/metadata/Metadata";  
import type...

REQ-3708: import { t } from "ttag";  
  
import {  
 isImplicitDeleteAction,  
 isImplicitUpdateAction,  
} from "metabase/actions/utils";  
import { formatValue, singularize } from "metabase/lib/formatting";  
import...

REQ-3709: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import Question from "metabase-lib/v1/Question";  
import {  
 createMockColumn,  
 createMockDatasetData,  
 createMockImplicitQueryAction,  
 ...

REQ-3710: import PinMap from "metabase/visualizations/components/PinMap";  
  
describe("PinMap", () => {  
 it("should filter out rows with null values in either the lat, long, or metric column", () => {  
 const...

REQ-3711: import { DEFAULT\_CARD\_SIZE, GRID\_WIDTH } from "metabase/lib/dashboard\_grid";  
import { measureText } from "metabase/lib/measure-text";  
  
interface FindSizeInput {  
 text: string;  
 targetHeight:...

REQ-3712: import \* as measureText from "metabase/lib/measure-text";  
import type { FontStyle } from "metabase/visualizations/shared/types/measure-text";  
  
import { findSize } from...

REQ-3713: import { useCallback, useMemo } from "react";  
  
import { useDispatch, useSelector } from "metabase/lib/redux";  
import { zoomInRow } from "metabase/query\_builder/actions";  
import { getRowIndexToPKMap }...

REQ-3714: import { useEffect, useRef } from "react";  
import \_ from "underscore";  
  
import type Question from "metabase-lib/v1/Question";  
import { isAdHocModelOrMetricQuestion } from...

REQ-3715: export \* from "./TableInteractive";

REQ-3716: export \* from "./TransformedVisualization";

REQ-3717: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3718: export { ErrorView } from "./ErrorView";

REQ-3719: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3720: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3721: import PropTypes from "prop-types";  
  
import { mockSettings } from "\_\_support\_\_/settings";  
import { renderWithProviders, screen } from "\_\_support\_\_/ui";  
import { delay } from...

REQ-3722: export \* from "./Watermark";

REQ-3723: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3724: export { LegendCaption } from "./LegendCaption";  
export {  
 LEGEND\_LABEL\_FONT\_SIZE,  
 LEGEND\_LABEL\_FONT\_WEIGHT,  
} from "./LegendCaption.styled";

REQ-3725: these tests use ChartSettings directly, but logic we're testing lives in ChartNestedSettingSeries

REQ-3726: export \* from "./ChartSettingColorPicker";

REQ-3727: these tests use QuestionChartSettings directly, but logic we're testing logic in ChartSettingFieldPicker

REQ-3728: import { render, screen } from "\_\_support\_\_/ui";  
import { createMockColumn } from "metabase-types/api/mocks";  
  
import { ChartSettingFieldsPartition } from...

REQ-3729: import userEvent from "@testing-library/user-event";  
  
import { renderWithProviders, screen } from "\_\_support\_\_/ui";  
import ChartSettingFieldsPicker from...

REQ-3730: export \* from "./ChartSettingOrderedItems";

REQ-3731: export \* from "./ChartSettingSegmentedControl";

REQ-3732: these tests use QuestionChartSettings directly, but logic we're testing lives in ChartNestedSettingSeries

REQ-3733: export \* from "./TableColumnPanel";

REQ-3734: import type { IconName } from "metabase/ui";  
import type {  
 DatasetColumn,  
 TableColumnOrderSetting,  
} from "metabase-types/api";  
  
export type ColumnItem = {  
 name: string;  
 enabled: boolean;  
 ...

REQ-3735: import \_ from "underscore";  
  
import type { ContentTranslationFunction } from "metabase/i18n/types";  
import type { IconName } from "metabase/ui";  
import { getIconForField } from...

REQ-3736: export \* from "./ChartSettingTableColumns";

REQ-3737: export type EditWidgetData = {  
 id: string;  
 props: EditWidgetProps;  
};  
  
export type EditWidgetProps = {  
 initialKey: string;  
};

REQ-3738: import \* as Lib from "metabase-lib";  
  
export function canEditQuery(query?: Lib.Query) {  
 if (!query) {  
 return false;  
 }  
 const { isNative, isEditable } = Lib.queryDisplayInfo(query);  
 return...

REQ-3739: import {  
 getAccentColors,  
 getStatusColorRanges,  
} from "metabase/lib/colors/groups";  
import type {  
 ColumnRangeFormattingSetting,  
 ColumnSingleFormattingSetting,  
} from...

REQ-3740: import { t } from "ttag";  
  
import {  
 isBoolean,  
 isFK,  
 isNumeric,  
 isPK,  
 isString,  
} from "metabase-lib/v1/types/utils/isa";  
import type {  
 ColumnFormattingOperator,  
 DatasetColumn,  
} from...

REQ-3741: import { createMockColumn } from "metabase-types/api/mocks";  
  
import { getOperatorsForColumns } from "./get-operators-for-columns";  
  
const STRING\_COLUMN = createMockColumn({  
 base\_type:...

REQ-3742: export { ChartSettingsTableFormatting } from "./ChartSettingsTableFormatting";  
export { isFormattable } from "./util";  
export { ALL\_OPERATOR\_NAMES } from "./get-operators-for-columns";

REQ-3743: import {  
 isBoolean,  
 isNumeric,  
 isString,  
} from "metabase-lib/v1/types/utils/isa";  
import type { DatasetColumn } from "metabase-types/api";  
  
// predicate for columns that can be...

REQ-3744: export \* from "./ColumnItem";

REQ-3745: import type { VisualizationSettings } from "metabase-types/api";  
  
export interface ChartSettingWidgetProps<TValue> {  
 value: TValue | undefined;  
 onChange: (value?: TValue | null) => void;  
 ...

REQ-3746: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3747: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3748: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3749: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3750: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3751: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3752: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3753: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3754: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3755: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3756: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3757: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3758: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3759: export type SkeletonCaptionSize = "medium" | "large";

REQ-3760: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3761: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3762: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3763: import \_ from "underscore";  
  
import { isNotNull } from "metabase/lib/types";  
import { X\_AXIS\_DATA\_KEY } from "metabase/visualizations/echarts/cartesian/constants/dataset";  
import { CHART\_STYLE } from...

REQ-3764: import type {  
 ComputedVisualizationSettings,  
 Padding,  
} from "metabase/visualizations/types";  
  
export interface TicksDimensions {  
 yTicksWidthLeft: number;  
 yTicksWidthRight: number;  
 ...

REQ-3765: We prefix misc data keys we add to avoid possible collisions with series data  
keys

REQ-3766: import type { LineSize } from "metabase-types/api";  
  
export const LINE\_SIZE: Record<LineSize, number> = {  
 S: 1,  
 M: 2,  
 L: 3,  
};  
  
export const Z\_INDEXES = {  
 // Note: timeline events use...

REQ-3767: import \* as d3 from "d3";  
import dayjs from "dayjs";  
import \_ from "underscore";  
  
import { NULL\_DISPLAY\_VALUE } from "metabase/lib/constants";  
import { formatValue } from...

REQ-3768: import { computeSplit } from "./axis";  
import type { SeriesExtents } from "./types";  
  
describe("computeSplit", () => {  
 const extents: SeriesExtents = {  
 "1": [6, 8],  
 "2": [9, 13],  
 "3":...

REQ-3769: import { t } from "ttag";  
  
import { getObjectKeys } from "metabase/lib/objects";  
import { parseTimestamp } from "metabase/lib/time-dayjs";  
import { checkNumber, isNotNull } from...

REQ-3770: import dayjs from "dayjs";  
  
import { createMockSeriesModel } from "\_\_support\_\_/echarts";  
import { checkNumber } from "metabase/lib/types";  
import {  
 ECHARTS\_CATEGORY\_AXIS\_NULL\_VALUE,  
 INDEX\_KEY,  
 ...

REQ-3771: import type {  
 BaseSeriesModel,  
 BreakoutSeriesModel,  
 CategoryXAxisModel,  
 NumericXAxisModel,  
 TimeSeriesInterval,  
 TimeSeriesXAxisModel,  
 XAxisModel,  
} from "./types";  
  
export const...

REQ-3772: import { OTHER\_DATA\_KEY } from "metabase/visualizations/echarts/cartesian/constants/dataset";  
import {  
 getXAxisModel,  
 getYAxesModels,  
} from...

REQ-3773: import { isBreakoutSeries } from "./guards";  
import type { BaseSeriesModel, LegendItem } from "./types";  
  
export const getLegendItems = (  
 seriesModels: BaseSeriesModel[],  
 showAllLegendItems:...

REQ-3774: import {  
 createMockBreakoutSeriesModel,  
 createMockSeriesModel,  
} from "\_\_support\_\_/echarts";  
  
import { getLegendItems } from "./legend";  
  
describe("getLegendItems", () => {  
 it("should return an...

REQ-3775: import { t } from "ttag";  
  
import { checkNumber } from "metabase/lib/types";  
import { isEmpty } from "metabase/lib/validate";  
import { SERIES\_SETTING\_KEY } from...

REQ-3776: import { memoize } from "metabase/common/hooks/use-memoized-callback";  
import { NULL\_DISPLAY\_VALUE } from "metabase/lib/constants";  
import { formatValue } from "metabase/lib/formatting";  
import type...

REQ-3777: import type {  
 BreakoutChartColumns,  
 CartesianChartColumns,  
} from "metabase/visualizations/lib/graph/columns";  
import { SERIES\_COLORS\_SETTING\_KEY } from...

REQ-3778: import \_ from "underscore";  
  
import { getObjectKeys } from "metabase/lib/objects";  
import type { ComputedVisualizationSettings } from "metabase/visualizations/types";  
  
import type { SeriesModel,...

REQ-3779: import { isNumber } from "metabase/lib/types";  
import type { NumericScale } from "metabase-types/api";  
  
import type { NumericAxisScaleTransforms } from "./types";  
  
function getSign(value: number) {  
 ...

REQ-3780: import Color from "color";  
  
import { checkNumber, isNotNull } from "metabase/lib/types";  
import { X\_AXIS\_DATA\_KEY } from "metabase/visualizations/echarts/cartesian/constants/dataset";  
import {...

REQ-3781: import type { Dayjs } from "dayjs";  
import type { OptionAxisType } from "echarts/types/src/coord/axisCommonTypes";  
  
import type {  
 INDEX\_KEY,  
 NEGATIVE\_STACK\_TOTAL\_DATA\_KEY,  
 ...

REQ-3782: import type { OptionsType } from "metabase/lib/formatting/types";  
import type {  
 ComputedVisualizationSettings,  
 RemappingHydratedDatasetColumn,  
} from "metabase/visualizations/types";  
import {...

REQ-3783: import type { XAXisOption, YAXisOption } from "echarts/types/dist/shared";  
import type { AxisBaseOptionCommon } from "echarts/types/src/coord/axisCommonTypes";  
  
import { parseNumberValue } from...

REQ-3784: import type { CustomSeriesOption } from "echarts/charts";  
  
import type {  
 ComputedVisualizationSettings,  
 RenderingContext,  
} from "metabase/visualizations/types";  
  
import type {...

REQ-3785: import type { EChartsCoreOption } from "echarts/core";  
import type { OptionSourceData } from "echarts/types/src/util/types";  
  
import {  
 NEGATIVE\_STACK\_TOTAL\_DATA\_KEY,  
 OTHER\_DATA\_KEY,  
 ...

REQ-3786: import type { XAXisOption, YAXisOption } from "echarts/types/dist/shared";  
  
import { DEFAULT\_VISUALIZATION\_THEME } from "metabase/visualizations/shared/utils/theme";  
import type { RenderingContext }...

REQ-3787: import type { BarSeriesOption, LineSeriesOption } from "echarts/charts";  
import type { CallbackDataParams } from "echarts/types/dist/shared";  
import type {  
 LabelLayoutOptionCallbackParams,  
 ...

REQ-3788: import type { Dayjs } from "dayjs";  
import dayjs from "dayjs";  
  
import type { ContinuousDomain } from "metabase/visualizations/shared/types/scale";  
  
import type {  
 TimeSeriesAxisFormatter,  
 ...

REQ-3789: import type { LineSeriesOption } from "echarts/charts";  
  
import { X\_AXIS\_DATA\_KEY } from "metabase/visualizations/echarts/cartesian/constants/dataset";  
  
import { Z\_INDEXES } from...

REQ-3790: import type {  
 BarSeriesOption,  
 CustomSeriesOption,  
 LineSeriesOption,  
 ScatterSeriesOption,  
} from "echarts/charts";  
  
export type EChartsSeriesOption =  
 | LineSeriesOption  
 | BarSeriesOption  
...

REQ-3791: import type { BaseCartesianChartModel, DataKey } from "../model/types";  
  
export function getSeriesYAxisIndex(  
 dataKey: DataKey,  
 chartModel: BaseCartesianChartModel,  
): number {  
 const {...

REQ-3792: import {  
 INDEX\_KEY,  
 X\_AXIS\_DATA\_KEY,  
} from "metabase/visualizations/echarts/cartesian/constants/dataset";  
import type { CartesianChartColumns } from...

REQ-3793: import { getObjectValues } from "metabase/lib/objects";  
import { isNotNull } from "metabase/lib/types";  
import type { ShowWarning } from "metabase/visualizations/echarts/types";  
import type {  
 ...

REQ-3794: import type { EChartsCoreOption } from "echarts/core";  
import type { OptionSourceData } from "echarts/types/src/util/types";  
  
import type {  
 ComputedVisualizationSettings,  
 RenderingContext,  
} from...

REQ-3795: import \* as d3 from "d3";  
import type { ScatterSeriesOption } from "echarts/charts";  
  
import { X\_AXIS\_DATA\_KEY } from "metabase/visualizations/echarts/cartesian/constants/dataset";  
import type {...

REQ-3796: import type { OpUnitType } from "dayjs";  
import dayjs from "dayjs";  
import \_ from "underscore";  
  
import { CHART\_STYLE } from "metabase/visualizations/echarts/cartesian/constants/style";  
import type...

REQ-3797: import type { TimeSeriesInterval } from "metabase/visualizations/echarts/cartesian/model/types";  
import type { TimelineEventGroup } from...

REQ-3798: import type { LineSeriesOption } from "echarts/charts";  
import type { MarkLine1DDataItemOption } from "echarts/types/src/component/marker/MarkLineModel";  
  
import type { IconName } from...

REQ-3799: import type { TimelineEvent } from "metabase-types/api";  
  
export type TimelineEventGroup = {  
 date: string;  
 events: TimelineEvent[];  
};  
  
export type TimelineEventsModel = TimelineEventGroup[];

REQ-3800: import type { Dayjs } from "dayjs";  
import dayjs from "dayjs";  
import \_ from "underscore";  
  
import { parseTimestamp } from "metabase/lib/time-dayjs";  
import { isNotNull } from...

REQ-3801: import dayjs from "dayjs";  
  
import testAcrossTimezones from "\_\_support\_\_/timezones";  
import { computeTimeseriesDataInterval } from...

REQ-3802: import dayjs from "dayjs";  
import timezone from "dayjs/plugin/timezone";  
import utc from "dayjs/plugin/utc";  
  
// Enable timezone and UTC plugins  
dayjs.extend(utc);  
dayjs.extend(timezone);  
  
import {...

REQ-3803: import { NULL\_CHAR } from "../constants/dataset";  
  
// Start of a waterfall bar  
export const WATERFALL\_START\_KEY = "start";  
// End of a waterfall bar  
export const WATERFALL\_END\_KEY = "end";  
// Value...

REQ-3804: import { t } from "ttag";  
  
import type {  
 ChartDataset,  
 DateRange,  
 DimensionModel,  
 Extent,  
 TimeSeriesXAxisModel,  
 WaterfallXAxisModel,  
} from...

REQ-3805: import { t } from "ttag";  
  
import { NULL\_DISPLAY\_VALUE } from "metabase/lib/constants";  
import {  
 INDEX\_KEY,  
 IS\_WATERFALL\_TOTAL\_DATA\_KEY,  
 X\_AXIS\_DATA\_KEY,  
} from...

REQ-3806: import { getYAxisModel } from "metabase/visualizations/echarts/cartesian/model/axis";  
import {  
 filterNullDimensionValues,  
 getCardsColumnByDataKeyMap,  
 getJoinedCardsDataset,  
 scaleDataset,  
 ...

REQ-3807: import type { EChartsCoreOption } from "echarts/core";  
import type { LabelLayoutOptionCallback } from "echarts/types/src/util/types";  
  
import { X\_AXIS\_DATA\_KEY } from...

REQ-3808: export const SANKEY\_CHART\_STYLE = {  
 nodeWidth: 20,  
 nodeLabels: {  
 weight: 400,  
 size: 13,  
 textBorderWidth: 3,  
 },  
 edgeLabels: {  
 weight: 400,  
 size: 12,  
 textBorderWidth:...

REQ-3809: import type {  
 ComputedVisualizationSettings,  
 RenderingContext,  
} from "metabase/visualizations/types";  
  
import { SANKEY\_CHART\_STYLE } from "../constants/style";  
import type { SankeyChartModel,...

REQ-3810: import type { Padding } from "metabase/visualizations/types";  
  
export type SankeyChartLayout = {  
 padding: Padding;  
 nodeIndicesWithTruncatedLabels: Set<number> | null;  
 truncateLabelWidth:...

REQ-3811: import { sumMetric } from "metabase/visualizations/lib/dataset";  
import { getColumnDescriptors } from "metabase/visualizations/lib/graph/columns";  
import type { ComputedVisualizationSettings } from...

REQ-3812: import { getColumnKey } from "metabase-lib/v1/queries/utils/column-key";  
import type { DatasetColumn } from "metabase-types/api";  
import { createMockCard } from...

REQ-3813: import { memoize } from "metabase/common/hooks/use-memoized-callback";  
import { formatValue } from "metabase/lib/formatting";  
import type { ComputedVisualizationSettings } from...

REQ-3814: import { t } from "ttag";  
  
import { getColorsForValues } from "metabase/lib/colors/charts";  
import type { ComputedVisualizationSettings } from "metabase/visualizations/types";  
import type { RawSeries...

REQ-3815: import type { ColumnDescriptor } from "metabase/visualizations/lib/graph/columns";  
import type { RowValue } from "metabase-types/api";  
  
export type ColumnKey = string;  
  
export interface...

REQ-3816: import type { SankeySeriesOption } from "echarts/charts";  
import type { EChartsCoreOption } from "echarts/core";  
  
import { truncateText } from "metabase/visualizations/lib/text";  
import type {  
 ...

REQ-3817: import {  
 BarChart,  
 CustomChart,  
 LineChart,  
 SankeyChart,  
 ScatterChart,  
 SunburstChart,  
} from "echarts/charts";  
import {  
 BrushComponent,  
 DataZoomComponent,  
 DatasetComponent,  
 ...

REQ-3818: import { t } from "ttag";  
  
import { NULL\_CHAR } from "../cartesian/constants/dataset";  
  
export const DIMENSIONS = {  
 maxSideLength: 550,  
 padding: {  
 legend: 16,  
 side: 12,  
 },  
 slice: {  
 ...

REQ-3819: import { NULL\_DISPLAY\_VALUE } from "metabase/lib/constants";  
import { formatValue } from "metabase/lib/formatting";  
import { computeMaxDecimalsForValues } from...

REQ-3820: import { pie } from "d3";  
import \_ from "underscore";  
  
import { findWithIndex } from "metabase/lib/arrays";  
import { checkNotNull } from "metabase/lib/types";  
import type { ColumnDescriptor } from...

REQ-3821: import type { RenderingContext } from "metabase/visualizations/types";  
import type { VisualizationSettings } from "metabase-types/api";  
import {  
 createMockCard,  
 createMockColumn,  
 ...

REQ-3822: import type { ColumnDescriptor } from "metabase/visualizations/lib/graph/columns";  
import type { RemappingHydratedDatasetColumn } from "metabase/visualizations/types";  
  
export interface PieRow {  
 ...

REQ-3823: import Color from "color";  
import type { EChartsOption, SunburstSeriesOption } from "echarts";  
  
import { getTextColorForBackground } from "metabase/lib/colors";  
import { checkNotNull } from...

REQ-3824: import type { EChartsSeriesMouseEvent } from "../types";  
  
type TreePathInfo = {  
 name: string;  
 dataIndex: number;  
 value: number;  
};  
  
export type EChartsSunburstSeriesMouseEvent =...

REQ-3825: import { match } from "ts-pattern";  
  
import { aliases, colors } from "metabase/lib/colors";  
import { isEmpty } from "metabase/lib/validate";  
  
const ACCENT\_KEY\_PREFIX = "accent";  
  
type AccentKey =...

REQ-3826: import { getPickerColorAlias, getRingColorAlias } from "./colors";  
  
describe("getRingColorAlias", () => {  
 it("should return the correct color alias for inner ring", () => {  
 ...

REQ-3827: import { checkNotNull } from "metabase/lib/types";  
  
import { OPTION\_NAME\_SEPERATOR } from "../constants";  
import type { SliceTree, SliceTreeNode } from "../model/types";  
import type {...

REQ-3828: export type Point = [number, number];  
  
/\*\*  
 \* Calculates the length of a chord given the radius of a circle and the central angle.  
 \* @param radius - The radius of the circle.  
 \* @param angle - The...

REQ-3829: import {  
 calcAvailableDonutSliceLabelLength,  
 calcChordLength,  
 calcCircleIntersectionByHorizontalLine,  
 calcInnerOuterRadiusesForRing,  
 getCoordOnCircle,  
} from...

REQ-3830: import { renderHook } from "@testing-library/react";  
import type { EChartsType } from "echarts/core";  
import type { MutableRefObject } from "react";  
  
import {  
 TOOLTIP\_POINTER\_MARGIN,  
 ...

REQ-3831: import type {  
 BarSeriesOption,  
 CustomSeriesOption,  
 LineSeriesOption,  
 ScatterSeriesOption,  
} from "echarts/charts";  
import type { ElementEvent } from "echarts/core";  
import type {...

REQ-3832: import { useMemo } from "react";  
  
import { usePalette } from "metabase/common/hooks/use-palette";  
import { getIsNightMode } from "metabase/dashboard/selectors";  
import { color } from...

REQ-3833: import { t } from "ttag";  
import \_ from "underscore";  
  
import { isStorybookActive } from "metabase/env";  
import type {  
 DatasetData,  
 RawSeries,  
 Series,  
 TransformedSeries,  
 ...

REQ-3834: import { push } from "react-router-redux";  
import \_ from "underscore";  
  
import { setParameterValuesFromQueryParams } from "metabase/dashboard/actions/parameters";  
import { isEmbeddingSdk } from...

REQ-3835: import MetabaseSettings from "metabase/lib/settings";  
import type {  
 QuestionChangeClickAction,  
 UrlClickAction,  
} from "metabase/visualizations/types";  
import Question from...

REQ-3836: import Color from "color";  
  
export function getHexColor(color: string) {  
 // Convert color values to hex format since Apache Batik (SVG renderer used in static visualizations)  
 // doesn't support...

REQ-3837: import { isNotNull } from "metabase/lib/types";  
import { isMetric } from "metabase-lib/v1/types/utils/isa";  
import type {  
 DatasetData,  
 RawSeries,  
 RowValue,  
 RowValues,  
} from...

REQ-3838: import {  
 createMockColumn,  
 createMockDatasetData,  
} from "metabase-types/api/mocks";  
  
import { groupDatasetMetrics, sumMetric } from "./dataset";  
  
describe("sumMetric", () => {  
 it("should...

REQ-3839: import { msgid, ngettext, t } from "ttag";  
  
export class MinColumnsError extends Error {  
 constructor(minColumns, actualColumns) {  
 super(  
 t`Doh! The data from your query doesn't fit the...

REQ-3840: import { MinRowsError } from "metabase/visualizations/lib/errors";  
  
describe("MinRowsError", () => {  
 it("should be an instanceof Error", () => {  
 expect(new MinRowsError(1, 0) instanceof...

REQ-3841: import {  
 createBrandingElement,  
 getBrandingConfig,  
 getBrandingSize,  
} from "./exports-branding-utils";  
  
describe("getBrandingSize", () => {  
 it("should return correct size for different...

REQ-3842: import { DASHBOARD\_HEADER\_PARAMETERS\_PDF\_EXPORT\_NODE\_ID } from "metabase/dashboard/constants";  
  
import { SAVING\_DOM\_IMAGE\_CLASS } from "./image-exports";  
const PARAMETERS\_MARGIN\_BOTTOM = 12;  
  
export...

REQ-3843: import \_ from "underscore";  
  
import { isNotNull } from "metabase/lib/types";  
import type { RemappingHydratedDatasetColumn } from "metabase/visualizations/types";  
import type {  
 DatasetColumn,  
 ...

REQ-3844: import { createMockColumn } from "metabase-types/api/mocks";  
  
import { getCartesianChartColumns } from "./columns";  
  
describe("getCartesianChartColumns", () => {  
 it("should ignore duplicated...

REQ-3845: eslint-disable-next-line no-restricted-imports

REQ-3846: import type { Dataset, JsonQuery } from "metabase-types/api";  
  
interface TileCoordinate {  
 x: number | string;  
 y: number | string;  
}  
  
interface TileUrlParams {  
 cardId?: number;  
 dashboardId?:...

REQ-3847: import type { JsonQuery } from "metabase-types/api";  
import { createMockDataset } from "metabase-types/api/mocks/dataset";  
  
import { getTileUrl } from "./map";  
  
describe("map", () => {  
 ...

REQ-3848: import \* as d3 from "d3";  
import L from "leaflet/dist/leaflet-src.js";  
  
import { COUNTRY\_NAME\_TO\_CODE, STATE\_CODES } from "./mapping\_codes";  
  
export function computeMinimalBounds(features) {  
 const...

REQ-3849: import { getCanonicalRowKey } from "./mapping";  
  
describe("getCanonicalRowKey", () => {  
 it("should convert US state names to their iso2 codes", () => {  
 expect(getCanonicalRowKey("Alabama",...

REQ-3850: export const STATE\_CODES = [  
 ["AL", "Alabama"],  
 ["AK", "Alaska"],  
 ["AS", "American Samoa"],  
 ["AZ", "Arizona"],  
 ["AR", "Arkansas"],  
 ["CA", "California"],  
 ["CO", "Colorado"],  
 ["CT",...

REQ-3851: import { isNumeric } from "metabase-lib/v1/types/utils/isa";  
  
export function dimensionIsNumeric({ cols, rows }, i = 0) {  
 if (isNumeric(cols[i])) {  
 return true;  
 }  
  
 const hasAtLeastOneNumber...

REQ-3852: import {  
 computeChange,  
 computeNumericDataInterval,  
 isMultipleOf,  
 precision,  
} from "metabase/visualizations/lib/numeric";  
  
describe("visualization.lib.numeric", () => {  
 ...

REQ-3853: import { getIn } from "icepick";  
import \_ from "underscore";  
  
import { formatNullable } from "metabase/lib/formatting/nullable";  
import { parseTimestamp } from "metabase/lib/time";  
import {...

REQ-3854: import moment from "moment-timezone"; // eslint-disable-line no-restricted-imports -- deprecated usage  
  
import {  
 getXValues,  
 parseXValue,  
} from...

REQ-3855: eslint-disable-next-line no-restricted-imports

REQ-3856: import { t } from "ttag";  
  
import { DASHBOARD\_HEADER\_PARAMETERS\_PDF\_EXPORT\_NODE\_ID } from "metabase/dashboard/constants";  
import type { Dashboard } from "metabase-types/api";  
  
import {  
 ...

REQ-3857: import { findPageBreakCandidates, getPageBreaks } from "./save-dashboard-pdf";  
  
describe("save-dashboard-pdf", () => {  
 describe("findPageBreakCandidates", () => {  
 it("should find a potential...

REQ-3858: import { formatValue } from "metabase/lib/formatting";  
import type { OptionsType } from "metabase/lib/formatting/types";  
import type { RowValue } from "metabase-types/api";  
  
export const...

REQ-3859: import { TYPE } from "metabase-lib/v1/types/constants";  
  
import {  
 COMPACT\_MAX\_WIDTH,  
 COMPACT\_MIN\_LENGTH,  
 COMPACT\_WIDTH\_PER\_DIGIT,  
 compactifyValue,  
} from "./scalar\_utils";  
  
describe("scalar...

REQ-3860: import { assocIn } from "icepick";  
  
import { SERIES\_SETTING\_KEY } from "metabase/visualizations/shared/settings/series";  
import type { Card, VisualizationSettings } from...

REQ-3861: import { trackSchemaEvent } from "metabase/lib/analytics";  
import type { DashboardId } from "metabase-types/api";  
  
export const trackCardSetToHideWhenNoResults = (dashboardId: DashboardId) => {  
 ...

REQ-3862: import { t } from "ttag";  
import \_ from "underscore";  
  
import { currency } from "cljs/metabase.util.currency";  
import {  
 displayNameForColumn,  
 getCurrency,  
 getCurrencyStyleOptions,  
 ...

REQ-3863: import { getComputedSettings } from "metabase/visualizations/lib/settings";  
import registerVisualizations from "metabase/visualizations/register";  
  
import { NUMBER\_COLUMN\_SETTINGS, columnSettings }...

REQ-3864: import { t } from "ttag";  
  
import { getDefaultGoalLabel } from "metabase/visualizations/shared/settings/cartesian-chart";  
import type { ChartGoal } from...

REQ-3865: import { t } from "ttag";  
import \_ from "underscore";  
  
import { color } from "metabase/lib/colors";  
import {  
 getMaxDimensionsSupported,  
 getMaxMetricsSupported,  
} from...

REQ-3866: import {  
 createMockCard,  
 createMockColumn,  
 createMockDataset,  
 createMockDatasetData,  
 createMockSingleSeries,  
} from "metabase-types/api/mocks";  
  
import {  
 GRAPH\_AXIS\_SETTINGS,  
 ...

REQ-3867: import { t } from "ttag";  
import \_ from "underscore";  
  
import chartSettingNestedSettings from "metabase/visualizations/components/settings/ChartSettingNestedSettings";  
  
import { getComputedSettings,...

REQ-3868: import { getComputedSettings } from "metabase/visualizations/lib/settings";  
import { nestedSettings } from "metabase/visualizations/lib/settings/nested";  
  
describe("nestedSettings", () => {  
 ...

REQ-3869: export const getNumberOr = <TValue, TReplacement>(  
 value: TValue,  
 replacement: TReplacement,  
): number | TReplacement => {  
 if (typeof value === "number") {  
 return value;  
 }  
  
 return...

REQ-3870: import { getIn } from "icepick";  
import { t } from "ttag";  
  
import ChartNestedSettingSeries from "metabase/visualizations/components/settings/ChartNestedSettingSeries";  
import { OTHER\_DATA\_KEY } from...

REQ-3871: import {  
 createMockCard,  
 createMockVisualizationSettings,  
} from "metabase-types/api/mocks";  
  
import { getColors } from "./series";  
  
describe("Series unit settings", () => {  
 ...

REQ-3872: import type { StackOffset } from "metabase/visualizations/shared/components/RowChart/types";  
import type { VisualizationSettings } from "metabase-types/api";  
  
export const getStackOffset = (  
 ...

REQ-3873: import { getIn } from "icepick";  
import \_ from "underscore";  
  
import { getVisualization } from "metabase/visualizations";  
import type { VisualizationSettingDefinition } from...

REQ-3874: import { registerVisualization } from "metabase/visualizations";  
import { BarChart } from "metabase/visualizations/visualizations/BarChart";  
import {  
 createMockCard,  
 ...

REQ-3875: import {  
 columnsAreValid,  
 getDefaultDimensionAndMetric,  
} from "metabase/visualizations/lib/utils";  
import { isDimension, isMetric } from "metabase-lib/v1/types/utils/isa";  
  
export function...

REQ-3876: import { t } from "ttag";  
import \_ from "underscore";  
  
import { isNotNull } from "metabase/lib/types";  
import {  
 ChartSettingsError,  
 MinRowsError,  
} from...

REQ-3877: import { assocIn } from "icepick";  
import { t } from "ttag";  
  
import { isVirtualDashCard } from "metabase/dashboard/utils";  
import { getVisualizationRaw } from "metabase/visualizations";  
import {...

REQ-3878: import icepick from "icepick";  
  
import { DateTimeColumn, NumberColumn } from "\_\_support\_\_/visualizations";  
import {  
 getComputedSettingsForSeries,  
 getStoredSettingsForSeries,  
} from...

REQ-3879: const PREFIX = "\0\_";  
  
// Encode boolean/null values to strings for Mantine form widgets (Select, Radio, etc), , needed for settings like "graph.x\_axis.axis\_enabled"  
const toWidgetValue = new...

REQ-3880: import \_ from "underscore";  
  
import { ChartSettingColorPicker } from "metabase/visualizations/components/settings/ChartSettingColorPicker";  
import ChartSettingColorsPicker from...

REQ-3881: NOTE: need to load visualizations first for getSettings to work

REQ-3882: import { isCoordinate, isNumber } from "metabase-lib/v1/types/utils/isa";  
  
/\*\*  
 \* @param {import("metabase-types/api").Series} series  
 \* @param {number} rowIndex  
 \* @param {number} columnIndex  
 \*...

REQ-3883: import {  
 getTableCellClickedObject,  
 getTableClickedObjectRowData,  
 isColumnRightAligned,  
} from "metabase/visualizations/lib/table";  
import { TYPE } from...

REQ-3884: NOTE: this file is used on the frontend and backend and there are some  
limitations. See frontend/src/metabase-shared/color\_selector for details

REQ-3885: import { ALL\_OPERATOR\_NAMES } from "metabase/visualizations/components/settings/ChartSettingsTableFormatting";  
import {  
 OPERATOR\_FORMATTER\_FACTORIES,  
 canCompareSubstrings,  
 compileFormatter,  
 ...

REQ-3886: import type {  
 FontStyle,  
 TextWidthMeasurer,  
} from "../shared/types/measure-text";  
  
export const CHAR\_ELLIPSES = "...";  
  
export function truncateText(  
 text: string,  
 width: number,  
 measurer:...

REQ-3887: import { measureTextWidth as measureDynamic } from "metabase/lib/measure-text";  
import { measureTextWidth as measureStatic } from "metabase/static-viz/lib/text";  
  
import type { TextWidthMeasurer }...

REQ-3888: import moment from "moment-timezone"; // eslint-disable-line no-restricted-imports -- deprecated usage  
  
import { isDate } from "metabase-lib/v1/types/utils/isa";  
  
const TIMESERIES\_UNITS = new Set([  
 ...

REQ-3889: import { dimensionIsTimeseries } from "metabase/visualizations/lib/timeseries";  
import registerVisualizations from "metabase/visualizations/register";  
import { TYPE } from...

REQ-3890: import { formatValue } from "metabase/lib/formatting";  
import { formatNullable } from "metabase/lib/formatting/nullable";  
import type { DatasetColumn, VisualizationSettings } from...

REQ-3891: mappings of allowed operators

REQ-3892: import crossfilter from "crossfilter";  
import \* as d3 from "d3";  
import \_ from "underscore";  
  
import { isNotNull } from "metabase/lib/types";  
import { getColumnKey } from...

REQ-3893: import \_ from "underscore";  
  
import {  
 cardHasBecomeDirty,  
 computeMaxDecimalsForValues,  
 computeSplit,  
 findSensibleSankeyColumns,  
 getCardAfterVisualizationClick,  
 getColumnCardinality,  
 ...

REQ-3894: import { t } from "ttag";  
  
const NULL\_DIMENSION\_WARNING = "NULL\_DIMENSION\_WARNING";  
export function nullDimensionWarning() {  
 return {  
 key: NULL\_DIMENSION\_WARNING,  
 text: t`Data includes...

REQ-3895: import ActionViz from "metabase/actions/components/ActionViz";  
import {  
 registerVisualization,  
 setDefaultVisualization,  
} from "metabase/visualizations";  
  
import { AreaChart } from...

REQ-3896: export const LABEL\_PADDING = 4;

REQ-3897: export \* from "./RowChart";

REQ-3898: import type { StringLike } from "@visx/scale";  
  
import type { AxisStyle, ChartFont, GoalStyle } from "../../types/style";  
  
export type XValue = number | null;  
export type YValue = string | number |...

REQ-3899: import type { Series as D3Series } from "d3";  
import \* as d3 from "d3";  
import { stack, stackOffsetDiverging, stackOffsetExpand } from "d3";  
import \_ from "underscore";  
  
import { formatNullable }...

REQ-3900: import { extent } from "d3";  
  
import { isNotNull } from "metabase/lib/types";  
import type {  
 ContinuousDomain,  
 ContinuousScaleType,  
} from "metabase/visualizations/shared/types/scale";  
  
import...

REQ-3901: import type { ScaleContinuousNumeric } from "d3-scale";  
  
import type { Margin } from "metabase/visualizations/shared/types/layout";  
import type { TextWidthMeasurer } from...

REQ-3902: import type { ContinuousDomain } from "@visx/scale";  
import { scaleBand, scaleLinear, scaleLog, scalePower } from "@visx/scale";  
import type { ScaleContinuousNumeric } from "d3-scale";  
  
import type {...

REQ-3903: import type { ScaleContinuousNumeric } from "d3-scale";  
import \_ from "underscore";  
  
import type { ValueFormatter } from "metabase/visualizations/shared/types/format";  
import type { TextWidthMeasurer...

REQ-3904: export const DATA\_LABEL\_OFFSET = 4;

REQ-3905: export \* from "./RowChartView";  
export \* from "./constants";

REQ-3906: import type { ScaleContinuousNumeric } from "d3-scale";  
  
import type { BarData } from "../../RowChart/types";  
  
export const getDataLabel = <TDatum>(  
 bar: BarData<TDatum>,  
 xScale:...

REQ-3907: import { scaleLinear } from "d3-scale";  
  
import type { BarData } from "../../RowChart/types";  
  
import { getDataLabel } from "./data-labels";  
  
describe("getDataLabel", () => {  
 const mockXScale =...

REQ-3908: import { t } from "ttag";  
import \_ from "underscore";  
  
import { isNotNull } from "metabase/lib/types";  
import {  
 getMaxDimensionsSupported,  
 getMaxMetricsSupported,  
} from...

REQ-3909: import { isCurrency, isPercentage } from "metabase-lib/v1/types/utils/isa";  
import type { ColumnSettings, DatasetColumn } from "metabase-types/api";  
  
export function getDefaultNumberStyle(  
 column:...

REQ-3910: import Color from "color";  
import \_ from "underscore";  
  
import { getColorsForValues } from "metabase/lib/colors/charts";  
import { NULL\_DISPLAY\_VALUE } from "metabase/lib/constants";  
import {...

REQ-3911: import { getIn } from "icepick";  
import \_ from "underscore";  
  
import { getColorsForValues } from "metabase/lib/colors/charts";  
import type { ComputedVisualizationSettings } from...

REQ-3912: import type { DatasetColumn, RowValue, RowValues } from "metabase-types/api";  
  
export type SeriesInfo = {  
 metricColumn: DatasetColumn;  
 dimensionColumn: DatasetColumn;  
 breakoutValue?:...

REQ-3913: export type HoveredData = {  
 seriesIndex: number;  
 datumIndex?: number;  
};

REQ-3914: import type { DatasetColumn } from "metabase-types/api";  
  
export type ValueFormatter = (value: any) => string;  
  
export type ColumnFormatter = (value: any, column: DatasetColumn) => string;  
  
export...

REQ-3915: export type Margin = {  
 top: number;  
 bottom: number;  
 right: number;  
 left: number;  
};

REQ-3916: export type FontStyle = {  
 size: string | number;  
 family: string;  
 weight: string | number;  
};  
  
export interface TextSize {  
 width: number;  
 height: number;  
}  
  
export type TextWidthMeasurer =...

REQ-3917: export type ContinuousScaleType = "linear" | "pow" | "log";  
export type ContinuousDomain = [number, number];  
export type Range = [number, number];

REQ-3918: export type ChartGoal = {  
 label: string;  
 value: number;  
};

REQ-3919: export type ChartFont = {  
 size: number;  
 family: string;  
 weight: number;  
 color: string;  
};  
  
export type GoalStyle = {  
 lineStroke: string;  
 label: ChartFont;  
};  
  
export type AxisStyle = {  
 ...

REQ-3920: import { getColorsForValues } from "metabase/lib/colors/charts";  
import type { VisualizationSettings } from "metabase-types/api";  
  
import type { Series } from "../components/RowChart/types";  
  
export...

REQ-3921: import { t } from "ttag";  
  
import { formatNullable } from "metabase/lib/formatting/nullable";  
import { getColumnScaling } from "metabase/visualizations/echarts/cartesian/model/util";  
import {...

REQ-3922: import type {  
 BreakoutChartColumns,  
 MultipleMetricsChartColumns,  
} from "metabase/visualizations/lib/graph/columns";  
import type { ColumnFormatter } from...

REQ-3923: import \_ from "underscore";  
  
import { substitute\_tags } from "cljs/metabase.parameters.shared";  
import { siteLocale, withInstanceLanguage } from "metabase/lib/i18n";  
  
export function...

REQ-3924: import type { CartesianChartColumns } from "metabase/visualizations/lib/graph/columns";  
import { getCartesianChartColumns } from "metabase/visualizations/lib/graph/columns";  
import type {...

REQ-3925: \*

REQ-3926: import { getSizeInPx } from "./size-in-px";  
  
describe("getSizeInPx", () => {  
 it("returns the number value if it's not a string", () => {  
 expect(getSizeInPx(12)).toBe(12);  
 });  
  
 it("returns...

REQ-3927: import \_ from "underscore";  
  
import { CARD\_SIZE\_DEFAULTS\_JSON } from "cljs/metabase.dashboards.constants";  
import { DEFAULT\_CARD\_SIZE } from "metabase/lib/dashboard\_grid";  
import type {...

REQ-3928: import { DEFAULT\_METABASE\_COMPONENT\_THEME } from "metabase/embedding-sdk/theme";  
import { color } from "metabase/lib/colors";  
import type { MantineThemeOther } from "metabase/ui";  
import {...

REQ-3929: import type React from "react";  
  
import type { IconName } from "metabase/ui";  
import type { Mode } from "metabase/visualizations/click-actions/Mode";  
import type \* as Lib from "metabase-lib";  
import...

REQ-3930: import type { DatasetColumn, DatasetData } from "metabase-types/api";  
  
export type RemappingHydratedDatasetColumn = DatasetColumn & {  
 remapped\_from\_index?: number;  
 remapped\_to\_column?:...

REQ-3931: export type EChartsEventHandler = {  
 eventName: string;  
 query?: string;  
 handler: (event: any) => void;  
};  
  
export type ZREventHandler = {  
 eventName: string;  
 handler: (event: any) => void;  
};

REQ-3932: import type { ClickObjectDataRow } from "metabase-lib";  
import type { RowValue, TimelineEvent } from "metabase-types/api";  
  
import type { RemappingHydratedDatasetColumn } from "./columns";  
import...

REQ-3933: export \* from "./hover";  
export \* from "./columns";  
export \* from "./visualization";  
export \* from "./click-actions";

REQ-3934: import type { ReactNode } from "react";  
import type React from "react";  
  
import type { OptionsType } from "metabase/lib/formatting/types";  
import type { IconName, IconProps } from...

REQ-3935: export \* from "./AreaChart";

REQ-3936: export \* from "./BarChart";

REQ-3937: import \_ from "underscore";  
  
import { NULL\_DISPLAY\_VALUE } from "metabase/lib/constants";  
import { formatValue } from "metabase/lib/formatting";  
import { isEmpty } from...

REQ-3938: import \_ from "underscore";  
  
import { GRAPH\_GOAL\_SETTINGS } from "metabase/visualizations/lib/settings/goal";  
import {  
 GRAPH\_AXIS\_SETTINGS,  
 GRAPH\_COLORS\_SETTINGS,  
 GRAPH\_DATA\_SETTINGS,  
 ...

REQ-3939: import { SERIES\_SETTING\_KEY } from "metabase/visualizations/shared/settings/series";  
  
import { getCartesianChartDefinition } from "./chart-definition";  
  
describe("chart-definition", () => {  
 ...

REQ-3940: import { t } from "ttag";  
import \_ from "underscore";  
  
import { NULL\_DISPLAY\_VALUE } from "metabase/lib/constants";  
import { formatChangeWithSign } from "metabase/lib/formatting";  
import {...

REQ-3941: export \* from "./CartesianChart";

REQ-3942: import type { MantineTheme } from "metabase/ui";  
  
export const getChartPadding = ({  
 theme,  
 isQueryBuilder,  
}: {  
 isQueryBuilder?: boolean;  
 theme: MantineTheme;  
}) => {  
 const { padding } =...

REQ-3943: eslint-disable no-console

REQ-3944: import type { EChartsCoreOption, EChartsType } from "echarts/core";  
import type \* as React from "react";  
import {  
 useCallback,  
 useEffect,  
 useLayoutEffect,  
 useMemo,  
 useRef,  
} from...

REQ-3945: import { useCallback, useMemo } from "react";  
  
import { useTranslateContent } from "metabase/i18n/hooks";  
import { isReducedMotionPreferred } from "metabase/lib/dom";  
import { extractRemappings }...

REQ-3946: import type { EChartsType } from "echarts/core";  
import { useEffect, useRef } from "react";  
import \_ from "underscore";  
  
import { ECHARTS\_TOOLTIP\_CONTAINER\_CLASS } from...

REQ-3947: import type { EChartsCoreOption } from "echarts/core";  
import { t } from "ttag";  
  
import { isNotNull } from "metabase/lib/types";  
import type {  
 BaseCartesianChartModel,  
 DataKey,  
 SeriesModel,  
}...

REQ-3948: export \* from "./ComboChart";

REQ-3949: import { formatValue } from "metabase/lib/formatting";  
import { isNotNull } from "metabase/lib/types";  
import type { TransformSeries } from...

REQ-3950: export \* from "./Funnel";

REQ-3951: import type { RowValue } from "metabase-types/api";  
  
export type FunnelRow = {  
 key: RowValue;  
 name: RowValue;  
 enabled: boolean;  
};

REQ-3952: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3953: export const getValue = (rows: unknown[][]) => {  
 const rawValue = rows[0] && rows[0][0];  
  
 if (rawValue === "Infinity") {  
 return Infinity;  
 }  
  
 if (typeof rawValue !== "number") {  
 return...

REQ-3954: import { getValue } from "./utils";  
  
describe("Visualizations > Gauge > utils", () => {  
 const valueTestCases = [  
 [[[null]], 0],  
 [[[undefined]], 0],  
 [[["foo"]], 0],  
 [[[""]], 0],  
 ...

REQ-3955: import { t } from "ttag";  
  
import {  
 getDefaultSize,  
 getMinSize,  
} from "metabase/visualizations/shared/utils/sizes";  
  
import { Heading } from "./Heading";  
  
const HeadingWrapper =...

REQ-3956: import { t } from "ttag";  
  
import {  
 getDefaultSize,  
 getMinSize,  
} from "metabase/visualizations/shared/utils/sizes";  
  
export const settings = {  
 getUiName: () => "iframe",  
 canSavePng: false,  
 ...

REQ-3957: export \* from "./IFrameViz";

REQ-3958: import { isSafeUrl } from "metabase/lib/formatting/link";  
  
/\*\*  
 \* Reconstructs a URL from its parts while preserving parameter placeholders (e.g. {{param}}).  
 \* Unlike URL.toString(), this won't...

REQ-3959: import {  
 getAllowedIframeAttributes,  
 getIframeDomainName,  
 isAllowedIframeUrl,  
} from "./utils";  
  
describe("getAllowedIframeAttributes", () => {  
 describe("share to embed link transformation",...

REQ-3960: export \* from "./LineChart";

REQ-3961: import { t } from "ttag";  
  
import {  
 getDefaultSize,  
 getMinSize,  
} from "metabase/visualizations/shared/utils/sizes";  
  
export const settings = {  
 getUiName: () => "Link",  
 canSavePng: false,  
 ...

REQ-3962: export { LinkViz } from "./LinkViz";

REQ-3963: import type { IconName } from "metabase/ui";  
import type { UnrestrictedLinkEntity } from "metabase-types/api";  
  
type WrappedEntity = {  
 getIcon: () => { name: IconName };  
 getUrl: () =>...

REQ-3964: export const isUrlString = (str?: string) => str && /^http/i.test(str);

REQ-3965: export { CustomMapFooter } from "./CustomMapFooter";

REQ-3966: import { screen } from "\_\_support\_\_/ui";  
  
import { setup } from "./setup";  
  
describe("CustomMapFooter (OSS)", () => {  
 describe("admin users", () => {  
 it("should show an admin settings link...

REQ-3967: import { screen } from "\_\_support\_\_/ui";  
  
import type { SetupOpts } from "./setup";  
import { setup as baseSetup } from "./setup";  
  
function setup(opts: SetupOpts) {  
 baseSetup({...

REQ-3968: import { screen } from "\_\_support\_\_/ui";  
  
import type { SetupOpts } from "./setup";  
import { setup as baseSetup } from "./setup";  
  
function setup(opts: SetupOpts) {  
 baseSetup({  
 ...

REQ-3969: export { Map } from "./Map";

REQ-3970: import { t } from "ttag";  
import \_ from "underscore";  
  
import { formatValue } from "metabase/lib/formatting";  
import {  
 ChartSettingsError,  
 MinRowsError,  
} from...

REQ-3971: export { PieChart } from "./PieChart";

REQ-3972: import type { EChartsType } from "echarts/core";  
import { type MutableRefObject, useEffect, useMemo } from "react";  
import { t } from "ttag";  
  
import { checkNotNull } from...

REQ-3973: import userEvent from "@testing-library/user-event";  
import { thaw } from "icepick";  
import { useState } from "react";  
  
import { createMockMetadata } from "\_\_support\_\_/metadata";  
import { render,...

REQ-3974: cell width and height for normal body cells

REQ-3975: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3976: import { getIn } from "icepick";  
import { t } from "ttag";  
import \_ from "underscore";  
  
import {  
 COLLAPSED\_ROWS\_SETTING,  
 COLUMN\_FORMATTING\_SETTING,  
 COLUMN\_SHOW\_TOTALS,  
 COLUMN\_SORT\_ORDER,  
 ...

REQ-3977: const cols = [  
 {  
 name: "field-123",  
 display\_name: "field-123",  
 source: "breakout",  
 field\_ref: ["field", 123, null],  
 },  
 {  
 name: "field-456",  
 display\_name: "field-456",  
 ...

REQ-3978: import type { ClickObjectDataRow, ClickObjectDimension } from "metabase-lib";  
import type { DatasetColumn } from "metabase-types/api";  
  
type PivotTableClickDimension = ClickObjectDimension & {  
 ...

REQ-3979: import { t } from "ttag";  
import \_ from "underscore";  
  
import { DEFAULT\_METABASE\_COMPONENT\_THEME } from "metabase/embedding-sdk/theme";  
import { sumArray } from "metabase/lib/arrays";  
import {...

REQ-3980: import type { PivotTableColumnSplitSetting } from "metabase-types/api";  
import { createMockColumn } from "metabase-types/api/mocks";  
  
import {  
 CELL\_PADDING,  
 MAX\_HEADER\_CELL\_WIDTH,  
 ...

REQ-3981: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3982: export const getValue = (rows: unknown[][]) => {  
 const rawValue = rows[0] && rows[0][0];  
  
 if (rawValue === "Infinity") {  
 return Infinity;  
 }  
  
 if (typeof rawValue !== "number") {  
 return...

REQ-3983: import { getValue } from "./utils";  
  
describe("Visualizations > Progress > utils", () => {  
 const valueTestCases = [  
 [[[null]], 0],  
 [[[undefined]], 0],  
 [[["foo"]], 0],  
 [[[""]], 0],  
 ...

REQ-3984: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-3985: import { getIn } from "icepick";  
import \_ from "underscore";  
  
import { formatNullable } from "metabase/lib/formatting/nullable";  
import { isNotNull } from "metabase/lib/types";  
import { sumMetric }...

REQ-3986: import type { MultipleMetricsChartColumns } from "metabase/visualizations/lib/graph/columns";  
import type {  
 BarData,  
 Series,  
} from...

REQ-3987: import type { NumberLike, StringLike } from "@visx/scale";  
  
import { NULL\_DISPLAY\_VALUE } from "metabase/lib/constants";  
import { formatValue } from "metabase/lib/formatting";  
import { isEmpty } from...

REQ-3988: import type { LegendItem } from "metabase/visualizations/echarts/cartesian/model/types";  
import type { Series } from "metabase/visualizations/shared/components/RowChart/types";  
import type {...

REQ-3989: import { t } from "ttag";  
  
import { GRAPH\_GOAL\_SETTINGS } from "metabase/visualizations/lib/settings/goal";  
import { getDefaultDimensionLabel } from...

REQ-3990: import { getStackOffset } from "metabase/visualizations/lib/settings/stacking";  
import type { Series } from "metabase/visualizations/shared/components/RowChart/types";  
import type { Range } from...

REQ-3991: import { useMemo } from "react";  
  
import { getIsNightMode } from "metabase/dashboard/selectors";  
import { color } from "metabase/lib/colors";  
import { useSelector } from "metabase/lib/redux";  
import...

REQ-3992: import type { CartesianChartColumns } from "metabase/visualizations/lib/graph/columns";  
import { unaggregatedDataWarning } from "metabase/visualizations/lib/warnings";  
import type { RowValues } from...

REQ-3993: import { t } from "ttag";  
  
import { getSankeyChartColumns } from "metabase/visualizations/echarts/graph/sankey/model/dataset";  
import { ChartSettingsError } from...

REQ-3994: import { ChartSettingsError } from "metabase/visualizations/lib/errors";  
import { createMockCard } from "metabase-types/api/mocks/card";  
import {  
 createMockColumn,  
 createMockDatasetData,  
} from...

REQ-3995: import type { EChartsType } from "echarts/core";  
import { useMemo } from "react";  
  
import type {  
 ColumnKey,  
 SankeyChartColumns,  
 SankeyLink,  
 SankeyNode,  
} from...

REQ-3996: import type { EChartsSeriesMouseEvent } from "metabase/visualizations/echarts/types";  
import { getColumnKey } from "metabase-lib/v1/queries/utils/column-key";  
import { createMockCard } from...

REQ-3997: export \* from "./SankeyChart";

REQ-3998: import sankeyDisconnectedGraphs from "./sankey-disconnected-graphs.json";  
import sankeyEdgeLabelsAuto from "./sankey-edge-labels-auto.json";  
import sankeyEdgeLabelsCompact from...

REQ-3999: import type { RowValue, RowValues } from "metabase-types/api";  
  
type Graph = Map<RowValue, Set<RowValue>>;  
  
const buildGraph = (  
 rows: RowValues[],  
 sourceIndex: number,  
 targetIndex: number,  
):...

REQ-4000: import { hasCyclicFlow } from "./cycle-detection";  
  
describe("hasCyclicFlow", () => {  
 it.each([  
 [  
 [  
 ["A", "B"],  
 ["B", "C"],  
 ["C", "A"],  
 ],  
 ],  
 [  
 ...

REQ-4001: export const PADDING = 32;  
  
export const TITLE\_ICON\_SIZE = 10;  
  
export const SCALAR\_TITLE\_LINE\_HEIGHT = 23;  
  
export const TITLE\_2\_LINES\_HEIGHT\_THRESHOLD = 120; // determined empirically

REQ-4002: export { Scalar } from "./Scalar";

REQ-4003: import { t } from "ttag";  
  
import type { TransformSeries } from "metabase/visualizations/components/TransformedVisualization";  
import { TYPE } from "metabase-lib/v1/types/constants";  
import type {...

REQ-4004: import {  
 PADDING,  
 SCALAR\_TITLE\_LINE\_HEIGHT,  
 TITLE\_2\_LINES\_HEIGHT\_THRESHOLD,  
 TITLE\_ICON\_SIZE,  
} from "./constants";  
  
export const getTitleLinesCount = (height: number) =>  
 height >...

REQ-4005: import { TITLE\_2\_LINES\_HEIGHT\_THRESHOLD } from "./constants";  
import { getValueHeight, getValueWidth } from "./utils";  
  
describe("Scalar > utils", () => {  
 describe("getValueHeight", () => {  
 ...

REQ-4006: export \* from "./ScatterPlot";

REQ-4007: import dayjs from "dayjs";  
import { t } from "ttag";  
import \_ from "underscore";  
  
import { formatValue } from "metabase/lib/formatting";  
import { formatDateTimeRangeWithUnit } from...

REQ-4008: import { color, colors } from "metabase/lib/colors";  
import { formatValue } from "metabase/lib/formatting/value";  
import { computeChange } from "metabase/visualizations/lib/numeric";  
import {  
 ...

REQ-4009: export const SPACING = 8;  
  
export const ICON\_SIZE = 13;  
  
export const TOOLTIP\_ICON\_SIZE = 11;  
  
export const ICON\_MARGIN\_RIGHT = SPACING;  
  
export const SCALAR\_TITLE\_LINE\_HEIGHT = 23;  
  
export const...

REQ-4010: export { SmartScalar } from "./SmartScalar";

REQ-4011: import { DateTimeColumn, NumberColumn } from "\_\_support\_\_/visualizations";  
import { COMPARISON\_TYPES } from "metabase/visualizations/visualizations/SmartScalar/constants";  
import type {  
 RowValues,  
...

REQ-4012: import type { COMPARISON\_TYPES } from "./constants";  
  
type AnotherColumnMenuOption = {  
 type: typeof COMPARISON\_TYPES.ANOTHER\_COLUMN;  
 name: string;  
};  
type PreviousValueMenuOption = {  
 type:...

REQ-4013: import dayjs from "dayjs";  
import { t } from "ttag";  
import \_ from "underscore";  
  
import { formatNumber } from "metabase/lib/formatting/numbers";  
import { measureText } from...

REQ-4014: import { DateTimeColumn, NumberColumn } from "\_\_support\_\_/visualizations";  
import \* as measureText from "metabase/lib/measure-text";  
import type { FontStyle } from...

REQ-4015: import userEvent from "@testing-library/user-event";  
import { thaw } from "icepick";  
import { useState } from "react";  
  
import { createMockMetadata } from "\_\_support\_\_/metadata";  
import {...

REQ-4016: import images from "./images.json";  
import ordersWithPeople from "./orders-with-people.json";  
import variousColumnSettings from "./various-column-settings.json";  
import wrappedLinks from...

REQ-4017: import { t } from "ttag";  
  
import {  
 getDefaultSize,  
 getMinSize,  
} from "metabase/visualizations/shared/utils/sizes";  
  
import { Text } from "./Text";  
  
const TextWrapper = Object.assign(Text, {  
 ...

REQ-4018: export \* from "./WaterfallChart";

REQ-4019: export \* from "./ColumnsListItem";  
export \* from "./ColumnsList";

REQ-4020: import type { ComputedVisualizationSettings } from "metabase/visualizations/types";  
import { groupColumnsBySuitableVizSettings } from "metabase/visualizer/visualizations/compat";  
import type {  
 ...

REQ-4021: import registerVisualizations from "metabase/visualizations/register";  
import type { Field } from "metabase-types/api";  
import {  
 createMockCategoryColumn,  
 createMockDataset,  
 ...

REQ-4022: export \* from "./DataImporter";

REQ-4023: export \* from "./DragOverlay";

REQ-4024: export { Footer } from "./Footer";

REQ-4025: export \* from "./Header";

REQ-4026: export \* from "./TabularPreviewModal";

REQ-4027: export \* from "./VisualizationCanvas";

REQ-4028: export \* from "./HorizontalWell";

REQ-4029: export \* from "./ScatterFloatingWell";

REQ-4030: export \* from "./VerticalWell";

REQ-4031: export \* from "./VisualizationPicker";

REQ-4032: export \* from "./Visualizer";

REQ-4033: export \* from "./VisualizerModal";

REQ-4034: export \* from "./VisualizerUiContext";

REQ-4035: export const DRAGGABLE\_ID = {  
 COLUMN: "COLUMN",  
 WELL\_ITEM: "WELL\_ITEM",  
};  
  
export const DROPPABLE\_ID = {  
 CANVAS\_MAIN: "CANVAS\_MAIN",  
  
 X\_AXIS\_WELL: "X\_AXIS\_WELL",  
 Y\_AXIS\_WELL:...

REQ-4036: import { useCallback, useState } from "react";  
  
/\*\*  
 \* A hook to manage one boolean state per key in an object.  
 \*/  
export const useBooleanMap = () => {  
 const [values, setValues] =...

REQ-4037: import type { Active } from "@dnd-kit/core";  
import { useMemo } from "react";  
  
import { useSelector } from "metabase/lib/redux";  
import {  
 getHoveredItems,  
 getReferencedColumns,  
} from...

REQ-4038: import { useMemo } from "react";  
import \_ from "underscore";  
  
import { useSelector } from "metabase/lib/redux";  
import type { VisualizerDataSource } from "metabase-types/api";  
  
import {  
 getCards,  
 ...

REQ-4039: import { useDispatch, useSelector } from "metabase/lib/redux";  
  
import { getCanRedo, getCanUndo } from "../selectors";  
import { redo, undo } from "../visualizer.slice";  
  
export function...

REQ-4040: import { createSelector } from "@reduxjs/toolkit";  
import \_ from "underscore";  
  
import {  
 extractRemappings,  
 getVisualization,  
 getVisualizationTransformed,  
 isCartesianChart,  
} from...

REQ-4041: import type { ClickObject } from "metabase/visualizations/types";  
import type {  
 DatasetColumn,  
 RawSeries,  
 VisualizerColumnValueSource,  
} from "metabase-types/api";  
  
import {...

REQ-4042: import \_ from "underscore";  
  
import { isPivotGroupColumn } from "metabase/lib/data\_grid";  
import { isDate, isDimension, isMetric } from "metabase-lib/v1/types/utils/isa";  
import type {  
 ...

REQ-4043: import visualizations from "metabase/visualizations";  
import type { DashboardCard, VisualizationDisplay } from "metabase-types/api";  
  
import { isVisualizerDashboardCard } from...

REQ-4044: import type {  
 VisualizerColumnValueSource,  
 VisualizerDataSource,  
 VisualizerDataSourceId,  
 VisualizerDataSourceNameReference,  
 VisualizerDataSourceType,  
} from "metabase-types/api";  
  
import {...

REQ-4045: import type { Active } from "@dnd-kit/core";  
  
import type {  
 DraggedColumn,  
 DraggedItem,  
 DraggedWellItem,  
} from "metabase-types/store/visualizer";  
  
import { DRAGGABLE\_ID } from...

REQ-4046: import { isPivotGroupColumn } from "metabase/lib/data\_grid";  
import { isNotNull } from "metabase/lib/types";  
import { isCartesianChart } from "metabase/visualizations";  
import {...

REQ-4047: import registerVisualizations from "metabase/visualizations/register";  
import type { CardDisplayType } from "metabase-types/api";  
import {  
 createMockCard,  
 createMockColumn,  
 ...

REQ-4048: import { isNotNull } from "metabase/lib/types";  
import type {  
 Card,  
 Dataset,  
 DatasetColumn,  
 RawSeries,  
 VisualizerColumnReference,  
 VisualizerDataSource,  
} from "metabase-types/api";  
import...

REQ-4049: import { registerVisualization } from "metabase/visualizations";  
import { LineChart } from "metabase/visualizations/visualizations/LineChart";  
import {  
 createMockCard,  
 createMockColumn,  
 ...

REQ-4050: import type {  
 Dataset,  
 VisualizerDashboardCard,  
 VisualizerDataSourceId,  
} from "metabase-types/api";  
  
import { createDataSource } from "./data-source";  
import { getVisualizationColumns } from...

REQ-4051: import \_ from "underscore";  
  
import { isNotNull } from "metabase/lib/types";  
import {  
 getMaxDimensionsSupported,  
 isCartesianChart,  
} from "metabase/visualizations";  
import type {  
 ...

REQ-4052: import registerVisualizations from "metabase/visualizations/register";  
import type { VisualizerColumnValueSource } from "metabase-types/api";  
import {  
 createMockCategoryColumn,  
 ...

REQ-4053: import type {  
 Dataset,  
 DatasetColumn,  
 VisualizerDataSource,  
 VisualizerDataSourceId,  
 VisualizerVizDefinition,  
} from "metabase-types/api";  
  
import {  
 createDimensionColumn,  
 ...

REQ-4054: import type {  
 Dataset,  
 VisualizerDataSource,  
 VisualizerVizDefinition,  
} from "metabase-types/api";  
import {  
 createMockColumn,  
 createMockDataset,  
 createMockDatasetData,  
} from...

REQ-4055: export \* from "./viz-settings";  
export \* from "./click-actions";  
export \* from "./column";  
export \* from "./dashboard-card-supports-visualizer";  
export \* from "./data-source";  
export \* from...

REQ-4056: import type {  
 BaseDashboardCard,  
 VisualizerDashboardCard,  
} from "metabase-types/api";  
  
export function isVisualizerDashboardCard(  
 dashcard?: BaseDashboardCard,  
): dashcard is...

REQ-4057: import \_ from "underscore";  
  
import type {  
 Dataset,  
 DatasetColumn,  
 RowValues,  
 VisualizerColumnValueSource,  
 VisualizerDataSource,  
 VisualizerDataSourceId,  
} from...

REQ-4058: import \_ from "underscore";  
  
import { isNotNull } from "metabase/lib/types";  
import { isCartesianChart } from "metabase/visualizations";  
import type {  
 RawSeries,  
 VisualizerColumnValueSource,  
}...

REQ-4059: import {  
 getColumnKey,  
 getColumnNameFromKey,  
} from "metabase-lib/v1/queries/utils/column-key";  
import type { VisualizationSettings } from "metabase-types/api";  
  
/\*\*  
 \* Recursively converts...

REQ-4060: import { getColumnKey } from "metabase-lib/v1/queries/utils/column-key";  
import type { VisualizationSettings } from "metabase-types/api";  
  
import {  
 updateVizSettingsKeysWithRefs,  
 ...

REQ-4061: import { getVisualization } from "metabase/visualizations";  
import type { VisualizationSettingDefinition } from "metabase/visualizations/types";  
import type { VisualizationDisplay } from...

REQ-4062: import { registerVisualization } from "metabase/visualizations";  
import { BarChart } from "metabase/visualizations/visualizations/BarChart";  
import { Map } from...

REQ-4063: import type { DragEndEvent } from "@dnd-kit/core";  
import type { Draft } from "immer";  
import \_ from "underscore";  
  
import { isNotNull } from "metabase/lib/types";  
import { isCartesianChart } from...

REQ-4064: import \_ from "underscore";  
  
import {  
 createMockCategoryColumn,  
 createMockColumn,  
 createMockDataset,  
 createMockDatetimeColumn,  
 createMockNumericColumn,  
} from...

REQ-4065: import \_ from "underscore";  
  
import { isCartesianChart } from "metabase/visualizations";  
import type { ComputedVisualizationSettings } from "metabase/visualizations/types";  
import type {  
 Dataset,  
 ...

REQ-4066: import type { DragEndEvent } from "@dnd-kit/core";  
import type { Draft } from "immer";  
import \_ from "underscore";  
  
import type { ComputedVisualizationSettings } from...

REQ-4067: import \_ from "underscore";  
  
import {  
 createMockCategoryColumn,  
 createMockDataset,  
 createMockNumericColumn,  
} from "metabase-types/api/mocks";  
import type { VisualizerVizDefinitionWithColumns }...

REQ-4068: import type { DragEndEvent } from "@dnd-kit/core";  
import \_ from "underscore";  
  
import { isNotNull } from "metabase/lib/types";  
import type { ComputedVisualizationSettings } from...

REQ-4069: import {  
 createMockCategoryColumn,  
 createMockDataset,  
 createMockNumericColumn,  
} from "metabase-types/api/mocks";  
import type { VisualizerVizDefinitionWithColumns } from...

REQ-4070: import type { VisualizerVizDefinitionWithColumns } from "metabase-types/store/visualizer";  
  
/\*\*  
 \* Ensures that the column is removed from the state if it is not used in any settings.  
 \*  
 \* @param...

REQ-4071: import type { DragEndEvent } from "@dnd-kit/core";  
import {  
 type PayloadAction,  
 createAction,  
 createSlice,  
} from "@reduxjs/toolkit";  
import { shallowEqual } from "react-redux";  
import...

REQ-4072: {  
 "rules": {  
 // Note: adding this rule to a eslint config file in a subfolder will remove  
 // \*not\* carry over the restricted imports from parent folders, you will  
 // need to copy them...

REQ-4073: import \* as ML from "cljs/metabase.lib.js";  
  
import { displayInfo } from "./metadata";  
import type {  
 Aggregable,  
 AggregationClause,  
 AggregationOperator,  
 ColumnMetadata,  
 Query,  
} from...

REQ-4074: import \* as ML from "cljs/metabase.lib.js";  
  
import { displayInfo } from "./metadata";  
import type { Bucket, Clause, ColumnMetadata, Query } from "./types";  
  
export function binning(clause: Clause |...

REQ-4075: import \* as ML from "cljs/metabase.lib.js";  
  
import { removeClause } from "./query";  
import type { BreakoutClause, ColumnMetadata, Query } from "./types";  
  
export function breakoutableColumns(  
 ...

REQ-4076: import { checkNotNull } from "metabase/lib/types";  
import \* as Lib from "metabase-lib";  
  
import { columnFinder, createQuery, findTemporalBucket } from "./test-helpers";  
  
describe("breakout", () => {  
...

REQ-4077: import { binning, isBinnable, withDefaultBinning } from "./binning";  
import {  
 isTemporalBucketable,  
 temporalBucket,  
 withDefaultTemporalBucket,  
} from "./temporal\_bucket";  
import type {...

REQ-4078: import \* as ML from "cljs/metabase.lib.js";  
import \* as TYPES from "cljs/metabase.lib.types.isa";  
import type Field from "metabase-lib/v1/metadata/Field";  
import type { Field as ApiField,...

REQ-4079: import \* as ML from "cljs/metabase.lib.js";  
import type { DatasetQuery, DimensionReference } from "metabase-types/api";  
  
import type { ColumnMetadata, Query } from "./types";  
  
export function...

REQ-4080: import \* as Lib from "metabase-lib";  
  
import { createQueryWithClauses } from "./test-helpers";  
  
describe("findColumnIndexesFromLegacyRefs", () => {  
 const stageIndex = -1;  
  
 it("should match...

REQ-4081: import \* as ML from "cljs/metabase.lib.js";  
  
import type { Query } from "./types";  
  
/\*\*  
 \* Get the Database ID (`:database`) associated with a query. If the query is using  
 \* the Saved Questions...

REQ-4082: import \* as ML from "cljs/metabase.lib.js";  
import type { CardId, DatasetColumn, RowValue } from "metabase-types/api";  
  
import type {  
 ClickObjectDataRow,  
 ClickObjectDimension,  
 ColumnMetadata,  
 ...

REQ-4083: import \* as ML from "cljs/metabase.lib.js";  
  
import type {  
 AggregationClause,  
 ColumnMetadata,  
 ExpressionArg,  
 ExpressionClause,  
 ExpressionOperator,  
 ExpressionOptions,  
 ExpressionParts,  
 ...

REQ-4084: import \* as ML from "cljs/metabase.lib.js";  
  
import { expressionParts } from "./expression";  
import type {  
 ColumnExtraction,  
 ColumnMetadata,  
 DrillThru,  
 ExpressionArg,  
 ExpressionClause,  
 ...

REQ-4085: import \* as ML from "cljs/metabase.lib.js";  
import type { FieldReference } from "metabase-types/api";  
  
import type {  
 Clause,  
 ColumnMetadata,  
 FieldValuesSearchInfo,  
 MetricMetadata,  
 Query,  
 ...

REQ-4086: import dayjs from "dayjs";  
import type { Moment } from "moment-timezone"; // eslint-disable-line no-restricted-imports -- deprecated usage  
  
import \* as ML from "cljs/metabase.lib.js";  
import type {...

REQ-4087: Note: only metabase-lib v2 exports should be added here

REQ-4088: import \* as ML from "cljs/metabase.lib.js";  
import type {  
 CardId,  
 ConcreteTableId,  
 DatabaseId,  
 VirtualTableId,  
} from "metabase-types/api";  
  
import type {  
 Bucket,  
 CardMetadata,  
 Clause,  
...

REQ-4089: import \* as ML from "cljs/metabase.lib.limit";  
  
import type { Limit, Query } from "./types";  
  
export function currentLimit(query: Query, stageIndex: number): Limit {  
 return ML.current\_limit(query,...

REQ-4090: import \* as ML from "cljs/metabase.lib.js";  
import \* as ML\_MetadataCalculation from "cljs/metabase.lib.metadata.calculation";  
import type {  
 CardId,  
 CardType,  
 DatabaseId,  
 DatasetColumn,  
 ...

REQ-4091: import \* as Lib from "metabase-lib";  
  
import { SAMPLE\_DATABASE, SAMPLE\_METADATA } from "./test-helpers";  
  
describe("metadataProvider", () => {  
 // this is a very important optimization that the FE...

REQ-4092: import \* as ML from "cljs/metabase.lib.js";  
  
import type { MetricMetadata, Query } from "./types";  
  
export function availableMetrics(  
 query: Query,  
 stageIndex: number,  
): MetricMetadata[] {  
 ...

REQ-4093: import \* as ML from "cljs/metabase.lib.js";  
import type { DatabaseId, TemplateTags } from "metabase-types/api";  
  
import type { MetadataProvider, Query } from "./types";  
  
export function nativeQuery(  
...

REQ-4094: import \* as Lib from "metabase-lib";  
  
import { SAMPLE\_DATABASE, SAMPLE\_METADATA } from "./test-helpers";  
  
describe("native query template tags", () => {  
 let metadataProvider:...

REQ-4095: import \* as ML from "cljs/metabase.lib.js";  
  
import { removeClause } from "./query";  
import type {  
 ColumnMetadata,  
 OrderByClause,  
 OrderByDirection,  
 Query,  
} from "./types";  
  
export function...

REQ-4096: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import \* as Lib from "metabase-lib";  
import { createMockCard } from "metabase-types/api/mocks";  
import {  
 SAMPLE\_DB\_ID,  
 ...

REQ-4097: import \* as ML from "cljs/metabase.lib.js";  
import type {  
 CardId,  
 CardType,  
 DatabaseId,  
 DatasetQuery,  
 TableId,  
} from "metabase-types/api";  
  
import type {  
 CardMetadata,  
 Clause,  
 ...

REQ-4098: import \* as Lib from "metabase-lib";  
  
import {  
 DEFAULT\_QUERY,  
 SAMPLE\_DATABASE,  
 SAMPLE\_METADATA,  
 createQuery,  
} from "./test-helpers";  
  
describe("fromLegacyQuery", () => {  
 // this is a very...

REQ-4099: import \* as ML from "cljs/metabase.lib.js";  
import type { SegmentId } from "metabase-types/api";  
  
import type { Query, SegmentMetadata } from "./types";  
  
export function availableSegments(  
 query:...

REQ-4100: import \* as ML from "cljs/metabase.lib.js";  
import type { TemporalUnit } from "metabase-types/api";  
  
import { displayInfo } from "./metadata";  
import type { Bucket, Clause, ColumnMetadata, Query }...

REQ-4101: import { describeTemporalInterval } from "./temporal\_bucket";  
  
describe("describeTemporalInterval", () => {  
 it("should return 'Previous 7 days' when include-current is false", () => {  
 const...

REQ-4102: istanbul ignore file

REQ-4103: import type { DefinedClauseName } from "metabase/querying/expressions";  
import type {  
 CardId,  
 DatabaseId,  
 DatasetColumn,  
 FieldId,  
 FieldValuesType,  
 RowValue,  
 SchemaId,  
 TableId,  
 ...

REQ-4104: export const ALERT\_TYPE\_ROWS = "alert-type-rows";  
export const ALERT\_TYPE\_TIMESERIES\_GOAL = "alert-type-timeseries-goal";  
export const ALERT\_TYPE\_PROGRESS\_BAR\_GOAL =...

REQ-4105: export {  
 ALERT\_TYPE\_ROWS,  
 ALERT\_TYPE\_TIMESERIES\_GOAL,  
 ALERT\_TYPE\_PROGRESS\_BAR\_GOAL,  
} from "./constants";

REQ-4106: import { t } from "ttag";  
  
import ValidationError, {  
 VALIDATION\_ERROR\_TYPES,  
} from "metabase-lib/v1/ValidationError";  
import type Metadata from "metabase-lib/v1/metadata/Metadata";  
import type...

REQ-4107: import type { TemplateTagDimension } from "metabase-lib/v1/Dimension";  
  
import type {  
 DimensionFK,  
 DimensionOptionsProps,  
 DimensionOptionsSection,  
} from "./types";  
  
// eslint-disable-next-line...

REQ-4108: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-4109: import type { TemplateTagDimension } from "metabase-lib/v1/Dimension";  
import type Field from "metabase-lib/v1/metadata/Field";  
  
interface DimensionOptionsSectionItem {  
 dimension:...

REQ-4110: eslint-disable-next-line @typescript-eslint/ban-ts-comment  
@ts-nocheck

REQ-4111: import type { ErrorType } from "./types";  
  
// eslint-disable-next-line import/no-default-export -- deprecated usage  
export default class ValidationError extends Error {  
 type?: ErrorType;  
  
 ...

REQ-4112: export const VALIDATION\_ERROR\_TYPES = {  
 MISSING\_TAG\_DIMENSION: "MISSING\_TAG\_DIMENSION",  
} as const;

REQ-4113: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-4114: import type { VALIDATION\_ERROR\_TYPES } from "./constants";  
  
export type ErrorType =  
 (typeof VALIDATION\_ERROR\_TYPES)[keyof typeof VALIDATION\_ERROR\_TYPES];

REQ-4115: import type Question from "metabase-lib/v1/Question";  
import type Database from "metabase-lib/v1/metadata/Database";  
import type { WritebackAction } from "metabase-types/api";  
  
export const...

REQ-4116: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import type { Database, WritebackAction } from "metabase-types/api";  
import {  
 createMockDatabase,  
 createMockQueryAction,  
} from...

REQ-4117: eslint-disable-next-line @typescript-eslint/ban-ts-comment  
@ts-nocheck  
eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-4118: eslint-disable-next-line @typescript-eslint/ban-ts-comment  
@ts-nocheck

REQ-4119: import \_ from "underscore";  
  
import { generateSchemaId } from "metabase-lib/v1/metadata/utils/schema";  
import type { NativeQuery, NormalizedDatabase } from "metabase-types/api";  
  
import Question from...

REQ-4120: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import NativeQuery from "metabase-lib/v1/queries/NativeQuery";  
import type { Database } from "metabase-types/api";  
import {...

REQ-4121: eslint-disable-next-line @typescript-eslint/ban-ts-comment  
@ts-nocheck

REQ-4122: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import { checkNotNull } from "metabase/lib/types";  
import { TYPE } from "metabase-lib/v1/types/constants";  
import type { Database, Field,...

REQ-4123: import type { NormalizedForeignKey } from "metabase-types/api";  
  
import type Field from "./Field";  
import type Metadata from "./Metadata";  
  
interface ForeignKey  
 extends Omit<NormalizedForeignKey,...

REQ-4124: import \_ from "underscore";  
  
import type {  
 CardId,  
 DatabaseId,  
 FieldId,  
 FieldReference,  
 SchemaId,  
 SegmentId,  
 SettingKey,  
 Settings,  
 TableId,  
} from "metabase-types/api";  
  
import type...

REQ-4125: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import {  
 createMockDatabase,  
 createMockField,  
 createMockSegment,  
 createMockTable,  
} from...

REQ-4126: import { humanize, titleize } from "metabase/lib/formatting";  
import type { NormalizedSchema } from "metabase-types/api";  
  
import type Database from "./Database";  
import type Metadata from...

REQ-4127: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import type { Table } from "metabase-types/api";  
import { createMockTable } from "metabase-types/api/mocks";  
  
const TEST\_TABLE =...

REQ-4128: import type { Filter, NormalizedSegment } from "metabase-types/api";  
  
import type Metadata from "./Metadata";  
import type Table from "./Table";  
  
interface Segment extends Omit<NormalizedSegment,...

REQ-4129: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import type { Segment } from "metabase-types/api";  
import { createMockSegment } from "metabase-types/api/mocks";  
  
interface SetupOpts {  
 ...

REQ-4130: import \_ from "underscore";  
  
// NOTE: this needs to be imported first due to some cyclical dependency nonsense  
import { singularize } from "metabase/lib/formatting";  
import type { NormalizedTable }...

REQ-4131: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import {  
 createMockField,  
 createMockForeignKey,  
 createMockTable,  
} from "metabase-types/api/mocks";  
  
const TABLE\_ORIGIN\_ID = 1;  
const...

REQ-4132: import { isVirtualCardId } from "metabase-lib/v1/metadata/utils/saved-questions";  
import {  
 BOOLEAN,  
 COORDINATE,  
 FOREIGN\_KEY,  
 LOCATION,  
 NUMBER,  
 PRIMARY\_KEY,  
 STRING,  
 STRING\_LIKE,  
 ...

REQ-4133: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import type { FieldReference } from "metabase-types/api";  
import { createMockField, createMockTable } from...

REQ-4134: import \* as Lib from "metabase-lib";  
import type Question from "metabase-lib/v1/Question";  
import type Database from "metabase-lib/v1/metadata/Database";  
import type NativeQuery from...

REQ-4135: import \_ from "underscore";  
  
import { createMockMetadata } from "\_\_support\_\_/metadata";  
import Question from "metabase-lib/v1/Question";  
import {  
 checkCanBeModel,  
 checkCanRefreshModelCache,  
 ...

REQ-4136: import { generateSchemaId } from "metabase-lib/v1/metadata/utils/schema";  
  
export const SAVED\_QUESTIONS\_VIRTUAL\_DB\_ID = -1337;  
const ROOT\_COLLECTION\_VIRTUAL\_SCHEMA\_NAME = "Everything else";  
  
export...

REQ-4137: import {  
 SAVED\_QUESTIONS\_VIRTUAL\_DB\_ID,  
 convertSavedQuestionToVirtualTable,  
 getCollectionVirtualSchemaId,  
 getCollectionVirtualSchemaName,  
 getQuestionIdFromVirtualTableId,  
 ...

REQ-4138: import type { DatabaseId, SchemaId, SchemaName } from "metabase-types/api";  
  
export const getSchemaName = (id: string | null | undefined): SchemaId => {  
 return parseSchemaId(id)[1];  
};  
  
type...

REQ-4139: import { generateSchemaId, getSchemaName, parseSchemaId } from "./schema";  
  
const SCHEMA\_TEST\_CASES = [  
 { dbId: 1, schemaName: 2, schema: "1:2" },  
 { dbId: 1, schemaName: "2", schema: "1:2" },  
 {...

REQ-4140: import { t } from "ttag";  
  
import {  
 BOOLEAN,  
 COORDINATE,  
 FOREIGN\_KEY,  
 LOCATION,  
 NUMBER,  
 PRIMARY\_KEY,  
 STRING,  
 STRING\_LIKE,  
 TEMPORAL,  
 TYPE,  
 UNKNOWN,  
} from...

REQ-4141: import \_ from "underscore";  
  
import {  
 FIELD\_FILTER\_OPERATORS,  
 FILTER\_OPERATORS\_BY\_TYPE\_ORDERED,  
} from "metabase-lib/v1/operators/constants";  
  
export function doesOperatorExist(operatorName) {  
 ...

REQ-4142: import {  
 doesOperatorExist,  
 getOperatorByTypeAndName,  
 isEqualsOperator,  
 isFuzzyOperator,  
} from "metabase-lib/v1/operators/utils/index";  
import {  
 COORDINATE,  
 FOREIGN\_KEY,  
 NUMBER,  
 ...

REQ-4143: import { t } from "ttag";  
  
export const PARAMETER\_OPERATOR\_TYPES = {  
 number: [  
 {  
 type: "number/=",  
 operator: "=",  
 get name() {  
 return t`Number`;  
 },  
 get...

REQ-4144: import type { UiParameter } from "metabase-lib/v1/parameters/types";  
  
export const createMockUiParameter = (  
 opts?: Partial<UiParameter>,  
): UiParameter => ({  
 id: "parameter-id",  
 slug: "slug",  
...

REQ-4145: import type Field from "metabase-lib/v1/metadata/Field";  
import type { Parameter, ParameterTarget } from "metabase-types/api";  
  
interface ValuePopulatedParameter extends...

REQ-4146: import Question from "metabase-lib/v1/Question";  
import type Metadata from "metabase-lib/v1/metadata/Metadata";  
import type {  
 ParameterWithTarget,  
 UiParameter,  
} from...

REQ-4147: import \_ from "underscore";  
  
import {  
 formatDateTimeForParameter,  
 formatDateToRangeForParameter,  
} from "metabase/lib/formatting/date";  
import type { ValueAndColumnForColumnNameDate } from...

REQ-4148: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import \* as dateFormatUtils from "metabase/lib/formatting/date";  
import { checkNotNull } from "metabase/lib/types";  
import Question from...

REQ-4149: import \* as Lib from "metabase-lib";  
import type { TemplateTagDimension } from "metabase-lib/v1/Dimension";  
import type Field from "metabase-lib/v1/metadata/Field";  
import { getParameterOperatorName...

REQ-4150: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import { TemplateTagDimension } from "metabase-lib/v1/Dimension";  
import Field from "metabase-lib/v1/metadata/Field";  
import type...

REQ-4151: import {  
 doesOperatorExist,  
 getOperatorByTypeAndName,  
} from "metabase-lib/v1/operators/utils";  
import { PARAMETER\_OPERATOR\_TYPES } from "metabase-lib/v1/parameters/constants";  
import {  
 ...

REQ-4152: import { createMockUiParameter } from "metabase-lib/v1/parameters/mock";  
  
import {  
 deriveFieldOperatorFromParameter,  
 getOperatorDisplayName,  
} from "./operators";  
  
const option = {  
 type:...

REQ-4153: import type Field from "metabase-lib/v1/metadata/Field";  
import type {  
 FieldFilterUiParameter,  
 UiParameter,  
} from "metabase-lib/v1/parameters/types";  
  
export const isFieldFilterUiParameter = (  
 ...

REQ-4154: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import type Field from "metabase-lib/v1/metadata/Field";  
import { createMockUiParameter } from "metabase-lib/v1/parameters/mock";  
import {...

REQ-4155: import type { Query } from "history";  
  
import {  
 normalizeBooleanParameterValue,  
 normalizeDateParameterValue,  
 normalizeNumberParameterValue,  
 normalizeStringParameterValue,  
 ...

REQ-4156: import { createMockParameter } from "metabase-types/api/mocks";  
  
import {  
 getParameterValueFromQueryParams,  
 getParameterValuesByIdFromQueryParams,  
} from...

REQ-4157: import type Field from "metabase-lib/v1/metadata/Field";  
import { isFuzzyOperator } from "metabase-lib/v1/operators/utils";  
import type {  
 Parameter,  
 ValuesQueryType,  
 ValuesSourceConfig,  
 ...

REQ-4158: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import type Field from "metabase-lib/v1/metadata/Field";  
import { createMockUiParameter } from "metabase-lib/v1/parameters/mock";  
import {...

REQ-4159: import \_ from "underscore";  
  
import { FIELD\_FILTER\_PARAMETER\_TYPES } from "metabase-lib/v1/parameters/constants";  
import type { Parameter, ParameterType } from "metabase-types/api";  
  
export function...

REQ-4160: import {  
 getParameterSubType,  
 getParameterType,  
} from "metabase-lib/v1/parameters/utils/parameter-type";  
  
describe("parameters/utils/parameter-type", () => {  
 describe("getParameterType", () =>...

REQ-4161: import \_ from "underscore";  
  
import {  
 getQueryType,  
 getSourceConfig,  
 getSourceType,  
} from "./parameter-source";  
import { getParameterType } from "./parameter-type";  
  
export const...

REQ-4162: import {  
 getParameterValue,  
 getParameterValuesBySlug,  
 getValuePopulatedParameters,  
 normalizeParameterValue,  
} from "metabase-lib/v1/parameters/utils/parameter-values";  
import {...

REQ-4163: import \_ from "underscore";  
  
import { isNotNull } from "metabase/lib/types";  
import \* as Lib from "metabase-lib";  
import type { TemplateTagDimension } from "metabase-lib/v1/Dimension";  
import type...

REQ-4164: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import { checkNotNull } from "metabase/lib/types";  
import \* as Lib from "metabase-lib";  
import {  
 columnFinder,  
 createQuery,  
 ...

REQ-4165: import { t } from "ttag";  
  
import type Field from "metabase-lib/v1/metadata/Field";  
import type { ParameterOptions, TemplateTag } from "metabase-types/api";  
import { createMockParameter } from...

REQ-4166: import \_ from "underscore";  
  
import { PARAMETER\_OPERATOR\_TYPES } from "metabase-lib/v1/parameters/constants";  
  
import {  
 getParameterOptions,  
 getParameterOptionsForField,  
} from...

REQ-4167: import \_ from "underscore";  
  
import type { ParameterWithTarget } from "metabase-lib/v1/parameters/types";  
import { getTemplateTagFromTarget } from "metabase-lib/v1/parameters/utils/targets";  
import...

REQ-4168: import {  
 getTemplateTagParameters,  
 getTemplateTags,  
 remapParameterValuesToTemplateTags,  
} from "metabase-lib/v1/parameters/utils/template-tags";  
import { createMockTemplateTag } from...

REQ-4169: import type { DatasetQuery } from "metabase-types/api";  
  
export class InternalQuery {  
 static isDatasetQueryType(datasetQuery: DatasetQuery) {  
 // eslint-disable-next-line...

REQ-4170: eslint-disable-next-line @typescript-eslint/ban-ts-comment  
@ts-nocheck

REQ-4171: export const STRUCTURED\_QUERY\_TEMPLATE = {  
 database: null,  
 type: "query",  
 query: {  
 "source-table": null,  
 },  
};

REQ-4172: import { getIn } from "icepick";  
import querystring from "querystring";  
import \_ from "underscore";  
  
import { renderLinkURLForClick } from "metabase/lib/formatting/link";  
import \* as Urls from...

REQ-4173: import \* as Lib from "metabase-lib";  
import type Question from "metabase-lib/v1/Question";  
  
interface FallbackNativeDrillProps {  
 question: Question;  
}  
  
export function nativeDrillFallback({...

REQ-4174: import type { ClickObject } from "metabase-lib";  
import type Question from "metabase-lib/v1/Question";  
import type { VisualizationSettings } from "metabase-types/api";  
  
export type {  
 ClickObject,  
 ...

REQ-4175: import { updateIn } from "icepick";  
import \_ from "underscore";  
  
import { copy } from "metabase/lib/utils";  
import \* as Lib from "metabase-lib";  
import { deriveFieldOperatorFromParameter } from...

REQ-4176: import {  
 createFieldReference,  
 getBaseDimensionReference,  
 getNormalizedDimensionReference,  
 hasStringFieldName,  
 isAggregationReference,  
 isExpressionReference,  
 isFieldReference,  
 ...

REQ-4177: import {  
 getColumnKey,  
 getColumnNameFromKey,  
 getColumnSettings,  
 getLegacyColumnKey,  
 getObjectColumnSettings,  
} from "metabase-lib/v1/queries/utils/column-key";  
import type { DatasetColumn }...

REQ-4178: import type { DatasetData, TableColumnOrderSetting } from "metabase-types/api";  
  
import type { DatasetColumnReference } from "./column-key";  
  
export const datasetContainsNoResults = (data:...

REQ-4179: import {  
 createMockColumn,  
 createMockTableColumnOrderSetting,  
} from "metabase-types/api/mocks";  
import { ORDERS } from "metabase-types/api/mocks/presets";  
  
import {  
 ...

REQ-4180: import { unique\_expression\_name } from "cljs/metabase.xrays.domain\_entities.queries.util";  
  
/\*\*  
 \* Ensures expression's name uniqueness  
 \*  
 \* Example: if query has a "Double Total" expression,  
 \* and...

REQ-4181: import \_ from "underscore";  
  
// Metadata field "values" type is inconsistent  
// https://github.com/metabase/metabase/issues/3417  
export function getFieldValues(field) {  
 const values = field &&...

REQ-4182: The backend won't return more than 2,000 rows so in cases where we  
need to communicate or use that, use this constant

REQ-4183: import \_ from "underscore";  
  
import type Table from "metabase-lib/v1/metadata/Table";  
import { getQuestionVirtualTableId } from "metabase-lib/v1/metadata/utils/saved-questions";  
  
import type...

REQ-4184: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import type Question from "metabase-lib/v1/Question";  
import type Table from "metabase-lib/v1/metadata/Table";  
import {...

REQ-4185: export { normalize } from "cljs/metabase.lib.js";

REQ-4186: import \_ from "underscore";  
  
import { isNotNull } from "metabase/lib/types";  
import \* as Lib from "metabase-lib";  
import type Question from "metabase-lib/v1/Question";  
import type {  
 ...

REQ-4187: import type { DatasetColumn } from "metabase-types/api";  
  
/\*\*  
 \* @returns min and max for a value in a column  
 \*/  
export const rangeForValue = (value: unknown, column: DatasetColumn) => {  
 if...

REQ-4188: import \_ from "underscore";  
  
import { normalize } from "metabase-lib/v1/queries/utils/normalize";  
import type {  
 AggregateFieldReference,  
 DimensionReference,  
 DimensionReferenceWithOptions,  
 ...

REQ-4189: import {  
 isAggregationReference,  
 isExpressionReference,  
 isFieldReference,  
 isTemplateTagReference,  
 normalizeReferenceOptions,  
} from "metabase-lib/v1/references";  
  
describe("reference...

REQ-4190: import {  
 LEVEL\_ONE\_TYPES as cljs\_LEVEL\_ONE\_TYPES,  
 TYPE as cljs\_TYPE,  
} from "cljs/metabase.types.core";  
  
export const LEVEL\_ONE\_TYPES: string[] = cljs\_LEVEL\_ONE\_TYPES;  
  
export const TYPE:...

REQ-4191: import { isa as cljs\_isa } from "cljs/metabase.types.core";  
import { isVirtualCardId } from "metabase-lib/v1/metadata/utils/saved-questions";  
import {  
 BOOLEAN,  
 COORDINATE,  
 FOREIGN\_KEY,  
 ...

REQ-4192: import {  
 BOOLEAN,  
 COORDINATE,  
 LOCATION,  
 NUMBER,  
 PRIMARY\_KEY,  
 STRING,  
 STRING\_LIKE,  
 TEMPORAL,  
 TYPE,  
} from "metabase-lib/v1/types/constants";  
import {  
 getFieldType,  
 isDimension,  
 ...

REQ-4193: import { utf8\_to\_b64url } from "metabase/lib/encoding";  
import \* as Urls from "metabase/lib/urls";  
import \* as Lib from "metabase-lib";  
import type { ParameterWithTarget } from...

REQ-4194: export { memoizeClass } from "./memoize-class";  
export { sortObject } from "./sort-object";

REQ-4195: type Constructor<T> = new (...args: any[]) => T;  
  
function getWithFallback(  
 map: Map<string, any>,  
 key: string,  
 fallback: () => void,  
) {  
 if (map.has(key)) {  
 return map.get(key);  
 } else...

REQ-4196: `sortObject` copies objects for deterministic serialization.  
Objects that have equal keys and values don't necessarily serialize to the  
same string. JSON.stringify prints properties in inserted...

REQ-4197: import NativeQuery from "metabase-lib/v1/queries/NativeQuery";  
import Variable from "metabase-lib/v1/variables/Variable";  
import type { TemplateTag, VariableTarget } from...

REQ-4198: TODO: migrate icons away from metabase-lib  
eslint-disable-next-line no-restricted-imports

REQ-4199: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-4200: import type Metadata from "metabase-lib/v1/metadata/Metadata";  
import type NativeQuery from "metabase-lib/v1/queries/NativeQuery";  
  
// eslint-disable-next-line import/no-default-export -- deprecated...

REQ-4201: eslint-disable-next-line import/no-default-export -- deprecated usage

REQ-4202: import \* as Lib from "metabase-lib";  
import type {  
 CardDisplayType,  
 VisualizationSettings,  
} from "metabase-types/api";  
  
type DefaultDisplay = {  
 display: CardDisplayType;  
 settings?:...

REQ-4203: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import type { Field, Table } from "metabase-types/api";  
import { createMockField, createMockTable } from "metabase-types/api/mocks";  
import...

REQ-4204: export \* from "./display";

REQ-4205: This runs in the Nashorn JavaScript engine and there are some limitations  
  
1. This is not currently automatically built with the rest of the application, please run `yarn build-shared` after...

REQ-4206: type AccountEventSchema = {  
 event: string;  
 version?: string | null;  
 link?: string | null;  
};  
  
type ValidateEvent<  
 T extends AccountEventSchema &  
 Record<Exclude<keyof T, keyof...

REQ-4207: type ActionEventSchema = {  
 event: string;  
 type: string;  
 action\_id: number;  
 num\_parameters?: number | null;  
 context?: string | null;  
};  
  
type ValidateEvent<  
 T extends ActionEventSchema &  
 ...

REQ-4208: type BrowseDataEventSchema = {  
 event: string;  
 model\_id?: number | null;  
 table\_id?: number | null;  
};  
  
type ValidateEvent<  
 T extends BrowseDataEventSchema &  
 Record<Exclude<keyof T, keyof...

REQ-4209: type CleanupEventSchema = {  
 event: string;  
 collection\_id?: number | null;  
 total\_stale\_items\_found?: number | null;  
 total\_items\_archived?: number | null;  
 cutoff\_date?: string |...

REQ-4210: type CsvUploadEventSchema = {  
 event: string;  
 model\_id?: number | null;  
 upload\_seconds?: number | null;  
 size\_mb: number | null;  
 num\_columns: number | null;  
 num\_rows: number | null;  
 ...

REQ-4211: type DashboardEventSchema = {  
 event: string;  
 dashboard\_id: number;  
 question\_id?: number | null;  
 num\_tabs?: number | null;  
 total\_num\_tabs?: number | null;  
 duration\_milliseconds?: number |...

REQ-4212: type DatabaseEventSchema = {  
 event: string;  
 database?: string | null;  
 database\_id?: number | null;  
 error\_type?: string | null;  
 source?: string | null;  
 dbms\_version?: string |...

REQ-4213: type DownloadsEventSchema = {  
 event: string;  
 resource\_type?: string | null;  
 accessed\_via?: string | null;  
 export\_type?: string | null;  
};  
  
type ValidateEvent<  
 T extends DownloadsEventSchema...

REQ-4214: import type { EmbedResourceDownloadOptions } from "metabase/public/lib/types";  
  
type EmbedFlowParams = {  
 locked?: number;  
 enabled?: number;  
 disabled?: number;  
};  
  
type EmbedFlowAppearance = {  
 ...

REQ-4215: type EmbedShareEventSchema = {  
 event: string;  
 authorized\_origins\_set?: boolean | null;  
 number\_embedded\_questions?: number | null;  
 number\_embedded\_dashboards?: number | null;  
};  
  
type...

REQ-4216: type EmbeddingHomepageEventSchema = {  
 event: string;  
 dismiss\_reason?: string | null;  
 initial\_tab?: string | null;  
};  
  
type ValidateEvent<  
 T extends EmbeddingHomepageEventSchema &  
 ...

REQ-4217: import type {  
 ChecklistItemCTA,  
 ChecklistItemValue,  
} from "metabase/home/components/Onboarding/types";  
import type { KeyboardShortcutId } from "metabase/palette/shortcuts";  
import type { Engine,...

REQ-4218: export \* from "./account";  
export \* from "./action";  
export \* from "./browse\_data";  
export \* from "./cleanup";  
export \* from "./csv-upload";  
export \* from "./dashboard";  
export \* from...

REQ-4219: type InviteEventSchema = {  
 event: string;  
 invited\_user\_id: number;  
 source?: string | null;  
};  
  
type ValidateEvent<  
 T extends InviteEventSchema &  
 Record<Exclude<keyof T, keyof...

REQ-4220: type ModelEventSchema = {  
 event: string;  
 model\_id: number;  
};  
  
type ValidateEvent<  
 T extends ModelEventSchema &  
 Record<Exclude<keyof T, keyof ModelEventSchema>, never>,  
> = T;  
  
export type...

REQ-4221: type QuestionEventSchema = {  
 event: string;  
 question\_id: number;  
 type?: string | null;  
 method?: string | null;  
 visualization\_type?: string | null;  
 database\_id?: number | null;  
 ...

REQ-4222: import type { AccountEvent } from "./account";  
import type { ActionEvent } from "./action";  
import type { BrowseDataEvent } from "./browse\_data";  
import type { CleanupEvent } from "./cleanup";  
import...

REQ-4223: type SearchEventSchema = {  
 event: string;  
 runtime\_milliseconds?: number | null;  
 context?: string | null;  
 total\_results?: number | null;  
 page\_results?: number | null;  
 position?: number |...

REQ-4224: type SerializationEventSchema = {  
 event: string;  
 source: string;  
 success: boolean;  
 direction?: string | null;  
 duration\_ms?: number | null;  
 error\_message?: string | null;  
 count?: number...

REQ-4225: type SettingsEventSchema = {  
 event: string;  
 source?: string | null;  
};  
  
type ValidateEvent<  
 T extends SettingsEventSchema &  
 Record<Exclude<keyof T, keyof SettingsEventSchema>, never>,  
> =...

REQ-4226: type SetupEventSchema = {  
 event: string;  
 version: string;  
 step?: string | null;  
 step\_number?: number | null;  
 usage\_reason?: string | null;  
 database?: string | null;  
 ...

REQ-4227: type TimelineEventSchema = {  
 event: string;  
 source?: string | null;  
 question\_id?: number | null;  
 collection\_id?: number | null;  
 time\_matters?: boolean | null;  
};  
  
type ValidateEvent<  
 T...

REQ-4228: type UpsellEventSchema = {  
 event: string;  
 promoted\_feature?: string | null;  
 upsell\_location?: string | null;  
};  
  
type ValidateEvent<  
 T extends UpsellEventSchema &  
 Record<Exclude<keyof T,...

REQ-4229: import type { CardId } from "./card";  
import type { DatabaseId } from "./database";  
import type { BaseEntityId } from "./entity-id";  
import type { Parameter, ParameterId, ParameterTarget } from...

REQ-4230: import type { CollectionId } from "./collection";  
import type { DashboardId } from "./dashboard";  
import type { DatabaseId, InitialSyncStatus } from "./database";  
import type { ModerationReviewStatus...

REQ-4231: export type ApiKeyId = number;  
  
export type ApiKey = {  
 name: string;  
 id: ApiKeyId;  
 group: {  
 id: number;  
 name: string;  
 };  
 creator\_id: number;  
 masked\_key: string;  
 created\_at:...

REQ-4232: export interface AIEntityAnalysisResponse {  
 summary: string;  
}  
  
export interface TimelineEventInfo {  
 name: string;  
 description?: string;  
 timestamp: string;  
}  
  
export interface...

REQ-4233: import type { DatasetQuery } from "metabase-types/api/query";  
  
export type FixSqlQueryRequest = {  
 query: DatasetQuery;  
 error\_message: string;  
};  
  
export type FixSqlQueryResponse = {  
 fixes:...

REQ-4234: import type { DatabaseId } from "metabase-types/api/database";  
  
export type GenerateSqlQueryRequest = {  
 prompt: string;  
 database\_id: DatabaseId;  
};  
  
export type GenerateSqlQueryResponse = {  
 ...

REQ-4235: import type { CardId } from "./card";  
import type { CollectionId } from "./collection";  
import type { DashCardId, DashboardId } from "./dashboard";  
import type { BaseEntityId } from...

REQ-4236: import type { CardType } from "./card";  
import type { DatabaseId } from "./database";  
  
export type AutocompleteRequest = {  
 databaseId: DatabaseId;  
 prefix?: string;  
 substring?:...

REQ-4237: import type { DatabaseId } from "metabase-types/api/database";  
  
export type XrayEntityType =  
 | "table"  
 | "segment"  
 | "metric"  
 | "model"  
 | "question"  
 | "adhoc"  
 | "field"  
 |...

REQ-4238: import type { CardId, CardType } from "./card";  
import type { CollectionId } from "./collection";  
import type { DashboardId } from "./dashboard";  
  
export const BOOKMARK\_TYPES = [  
 "card",  
 ...

REQ-4239: import type {  
 Card,  
 Collection,  
 Dashboard,  
 DatasetData,  
 Log,  
 MetabaseInfo,  
} from "metabase-types/api";  
  
export type ReportableEntityName =  
 | "question"  
 | "model"  
 | "dashboard"  
 |...

REQ-4240: import type { EmbeddingParameters } from "metabase/public/lib/types";  
import type { PieRow } from "metabase/visualizations/echarts/pie/model/types";  
  
import type { Collection, CollectionId,...

REQ-4241: import type {  
 CardId,  
 DashboardId,  
 DashboardTabId,  
 DimensionTargetOptions,  
 FieldReference,  
 ParameterId,  
 TemplateTagName,  
} from "metabase-types/api";  
  
// Used to set values for question...

REQ-4242: export type CloudMigrationState =  
 | "init"  
 | "setup"  
 | "dump"  
 | "upload"  
 | "cancelled"  
 | "error"  
 | "done";  
  
export type CloudMigration = {  
 id: number;  
 external\_id: string;  
 state:...

REQ-4243: import type { ColorName } from "metabase/lib/colors/types";  
import type { IconName, IconProps } from "metabase/ui";  
import type {  
 BaseEntityId,  
 CollectionEssentials,  
 Dashboard,  
 DashboardId,  
 ...

REQ-4244: export type DictionaryArrayRow = {  
 locale: string;  
 msgid: string;  
 msgstr: string;  
};  
  
export type DictionaryArray = DictionaryArrayRow[];  
  
/\*\* Translations retrieved from the BE have ids...

REQ-4245: import type { EmbeddingParameters } from "metabase/public/lib/types";  
import type {  
 BaseEntityId,  
 CardDisplayType,  
 ClickBehavior,  
 Collection,  
 CollectionAuthorityLevel,  
 CollectionId,  
 ...

REQ-4246: import type { ScheduleSettings } from "./settings";  
import type { Table } from "./table";  
  
import type { ISO8601Time, LongTaskStatus } from ".";  
  
export type DatabaseId = number;  
  
export type...

REQ-4247: import type {  
 CacheStrategy,  
 LocalFieldReference,  
 Parameter,  
 ParameterValueOrArray,  
 VisualizerColumnValueSource,  
} from "metabase-types/api";  
  
import type { Card } from "./card";  
import...

REQ-4248: export interface EmailSMTPSettings {  
 "email-smtp-host": string;  
 "email-smtp-password": string | null;  
 "email-smtp-port": number | null;  
 "email-smtp-security": "none" | "ssl" | "tls" |...

REQ-4249: /\*  
BaseEntityId should NOT be created on the frontend - these IDs are generated by the backend when an entity  
is created. As FE developers we should not be touching or modifying these IDs without any...

REQ-4250: import type { RowValue } from "./dataset";  
import type { FieldReference } from "./query";  
import type { Table, TableId } from "./table";  
  
export type FieldId = number;  
  
export interface...

REQ-4251: import type { Feature, FeatureCollection } from "geojson";  
  
export type GeoJSONData = Feature | FeatureCollection;

REQ-4252: import type { User } from "./user";  
  
export type GroupId = number;  
  
export type Membership = {  
 user\_id: number;  
 group\_id: number;  
 membership\_id: number;  
 is\_group\_manager?: boolean;  
};  
  
export...

REQ-4253: export \* from "./actions";  
export \* from "./activity";  
export \* from "./admin";  
export \* from "./ai-sql-fixer";  
export \* from "./ai-sql-generation";  
export \* from "./ai-entity-analysis";  
export \*...

REQ-4254: import type { DateTimeAbsoluteUnit } from "./query";  
  
export type InsightExpressionOperator =  
 | "+"  
 | "-"  
 | "\*"  
 | "/"  
 | "log"  
 | "pow"  
 | "exp";  
  
export type InsightExpressionOperand =...

REQ-4255: export type LogLevel =  
 | "off"  
 | "fatal"  
 | "error"  
 | "warn"  
 | "info"  
 | "debug"  
 | "trace";  
  
export type LoggerDurationUnit =  
 | "days"  
 | "hours"  
 | "minutes"  
 | "seconds"  
 |...

REQ-4256: import type {  
 CardDisplayType,  
 CardId,  
 CardType,  
 CollectionId,  
 DashboardId,  
 DatasetQuery,  
 PaginationResponse,  
 RowValue,  
 SearchModel,  
} from ".";  
  
export type MetabotFeedbackType =  
 ...

REQ-4257: import type {  
 ActionFormSettings,  
 CardId,  
 FieldSettings,  
 PublicWritebackAction,  
 WritebackImplicitQueryAction,  
 WritebackParameter,  
 WritebackQueryAction,  
} from...

REQ-4258: import type {  
 PopularItem,  
 RecentCollectionItem,  
 RecentTableDatabaseInfo,  
 RecentTableItem,  
} from "metabase-types/api";  
  
export const createMockRecentTableItem = (  
 opts?:...

REQ-4259: import type {  
 FixSqlQueryRequest,  
 FixSqlQueryResponse,  
 SqlQueryFix,  
} from "metabase-types/api";  
  
import { createMockNativeDatasetQuery } from "./query";  
  
export function...

REQ-4260: import type {  
 GenerateSqlQueryRequest,  
 GenerateSqlQueryResponse,  
} from "metabase-types/api";  
  
export const createMockGenerateSqlQueryRequest = (  
 opts?: Partial<GenerateSqlQueryRequest>,  
):...

REQ-4261: import type { Alert, AlertCard } from "../alert";  
import type { Channel } from "../notification-channels";  
  
import { createMockEntityId } from "./entity-id";  
import { createMockUserInfo } from...

REQ-4262: import type { DatabaseXray, TableXray } from "metabase-types/api";  
  
export const createMockDatabaseCandidate = (  
 opts?: Partial<DatabaseXray>,  
): DatabaseXray => ({  
 id: "1/public",  
 schema:...

REQ-4263: import type { Bookmark } from "metabase-types/api";  
  
export const createMockBookmark = (opts?: Partial<Bookmark>): Bookmark => ({  
 id: "collection-1",  
 name: "My Collection",  
 item\_id: 1,  
 type:...

REQ-4264: import type {  
 Card,  
 CardQueryMetadata,  
 ColumnRangeFormattingSetting,  
 ColumnSingleFormattingSetting,  
 ModerationReview,  
 NativeDatasetQuery,  
 PublicCard,  
 SeriesOrderSetting,  
 ...

REQ-4265: import type {  
 ChannelDetails,  
 NotificationChannel,  
} from "metabase-types/api/notification-channels";  
  
export const createMockChannelDetails = (  
 opts: Partial<ChannelDetails>,  
): ChannelDetails...

REQ-4266: import type {  
 BaseEntityId,  
 Collection,  
 CollectionEssentials,  
 CollectionItem,  
} from "metabase-types/api";  
  
import { createMockEntityId } from "./entity-id";  
  
export const...

REQ-4267: import type {  
 ActionDashboardCard,  
 Dashboard,  
 DashboardQueryMetadata,  
 DashboardTab,  
 QuestionDashboardCard,  
 VirtualCard,  
 VirtualDashboardCard,  
} from "metabase-types/api";  
  
import {...

REQ-4268: import type {  
 Database,  
 DatabaseData,  
 DatabaseFeature,  
 SavedQuestionDatabase,  
} from "metabase-types/api";  
  
export const COMMON\_DATABASE\_FEATURES: DatabaseFeature[] = [  
 "actions",  
 ...

REQ-4269: import type {  
 Dataset,  
 DatasetColumn,  
 DatasetData,  
 EmbedDataset,  
 EmbedDatasetData,  
 ErrorEmbedDataset,  
 ResultsMetadata,  
 TemplateTag,  
} from "metabase-types/api/dataset";  
  
export const...

REQ-4270: import { nanoid } from "@reduxjs/toolkit";  
  
import { type BaseEntityId, NANOID\_LENGTH } from "../entity-id";  
  
export const createMockEntityId = (value?: string): BaseEntityId =>  
 (value ||...

REQ-4271: import type {  
 DateTimeFieldFingerprint,  
 Field,  
 FieldDimension,  
 FieldFingerprint,  
 FieldGlobalFingerprint,  
 GetFieldValuesResponse,  
 NumberFieldFingerprint,  
 TextFieldFingerprint,  
} from...

REQ-4272: import type { FeatureCollection } from "geojson";  
  
export const createMockGeoJSONFeatureCollection = (  
 opts?: Partial<FeatureCollection>,  
): FeatureCollection => {  
 return {  
 type:...

REQ-4273: import type { Group } from "metabase-types/api";  
  
export const createMockGroup = (  
 opts?: Partial<Group>,  
): Omit<Group, "members"> => ({  
 id: 1,  
 name: "All Users",  
 member\_count: 1,  
 ...

REQ-4274: export \* from "./actions";  
export \* from "./activity";  
export \* from "./ai-sql-fixer";  
export \* from "./ai-sql-generation";  
export \* from "./alert";  
export \* from "./automagic-dashboards";  
export \*...

REQ-4275: import type { LoggerPreset } from "../logger";  
  
export function createMockLoggerPreset(  
 preset?: Partial<LoggerPreset>,  
): LoggerPreset {  
 return {  
 display\_name: "Logger preset",  
 id: "1",  
...

REQ-4276: import type { ModelIndex } from "metabase-types/api";  
  
export const createMockModelIndex = (  
 opts?: Partial<ModelIndex>,  
): ModelIndex => ({  
 id: 1,  
 model\_id: 1,  
 pk\_ref: ["field", 1, null],  
 ...

REQ-4277: import type { ModelCacheRefreshStatus } from "metabase-types/api";  
  
export const getMockModelCacheInfo = (  
 opts?: Partial<ModelCacheRefreshStatus>,  
): ModelCacheRefreshStatus => {  
 const now = new...

REQ-4278: import type {  
 Notification,  
 NotificationCronSubscription,  
 NotificationHandlerEmail,  
 NotificationHandlerSlack,  
 NotificationRecipientUser,  
} from "metabase-types/api";  
import {...

REQ-4279: import type {  
 Parameter,  
 ParameterValues,  
 SearchParameterValuesRequest,  
} from "metabase-types/api";  
  
export const createMockParameter = (opts?: Partial<Parameter>): Parameter => ({  
 id: "1",  
...

REQ-4280: import { type CacheConfig, CacheDurationUnit } from "metabase-types/api";  
  
export const createMockCacheConfig = (  
 opts?: Partial<CacheConfig>,  
): CacheConfig => ({  
 model: "database",  
 model\_id:...

REQ-4281: import {  
 DataPermission,  
 DataPermissionValue,  
} from "metabase/admin/permissions/types";  
import type {  
 Database,  
 Group,  
 GroupsPermissions,  
 Impersonation,  
 PermissionsGraph,  
} from...

REQ-4282: export \* from "./sample\_cards";  
export \* from "./sample\_database";

REQ-4283: import type {  
 Card,  
 NativeDatasetQuery,  
 StructuredDatasetQuery,  
 UnsavedCard,  
} from "metabase-types/api";  
import {  
 createMockNativeCard,  
 createMockStructuredCard,  
} from...

REQ-4284: import type {  
 Database,  
 DatasetColumn,  
 Field,  
 FieldDimensionOption,  
 GetFieldValuesResponse,  
 Table,  
} from "metabase-types/api";  
import {  
 createMockColumn,  
 createMockDatabase,  
 ...

REQ-4285: import type { Pulse } from "metabase-types/api";  
  
export const createMockPulse = (opts?: Partial<Pulse>): Pulse => ({  
 name: "Pulse",  
 cards: [],  
 channels: [],  
 parameters: [],  
 ...opts,  
});

REQ-4286: import type {  
 NativeDatasetQuery,  
 NativeQuery,  
 StructuredDatasetQuery,  
 StructuredQuery,  
} from "metabase-types/api";  
  
export const createMockStructuredQuery = (  
 opts?:...

REQ-4287: import type { Revision } from "metabase-types/api";  
  
export const createMockRevision = (opts?: Partial<Revision>): Revision => ({  
 model\_id: 1,  
 id: 1,  
 description: "created this",  
 message:...

REQ-4288: import type {  
 NormalizedCollection,  
 NormalizedDatabase,  
 NormalizedField,  
 NormalizedFieldDimension,  
 NormalizedSchema,  
 NormalizedSegment,  
 NormalizedTable,  
 NormalizedTimeline,  
} from...

REQ-4289: import \_ from "underscore";  
  
import type { SearchResponse, SearchResult } from "metabase-types/api";  
  
import { createMockCollection } from "./collection";  
  
export const createMockSearchResult = (  
 ...

REQ-4290: import type { Segment } from "metabase-types/api";  
  
import { createMockStructuredQuery } from "./query";  
  
export const createMockSegment = (opts?: Partial<Segment>): Segment => ({  
 id: 1,  
 name:...

REQ-4291: import type { Card } from "../card";  
import type { Series, SingleSeries } from "../dataset";  
  
import { createMockCard } from "./card";  
import type { MockDatasetOpts } from "./dataset";  
import {...

REQ-4292: import type { PasswordResetTokenStatus } from "metabase-types/api";  
  
export const createMockPasswordResetTokenStatus = (  
 opts?: Partial<PasswordResetTokenStatus>,  
): PasswordResetTokenStatus => ({  
...

REQ-4293: import dayjs from "dayjs";  
  
import type {  
 Engine,  
 EngineField,  
 EngineSource,  
 EnterpriseSettingKey,  
 EnterpriseSettings,  
 FontFile,  
 SettingDefinition,  
 Settings,  
 TokenFeatures,  
 ...

REQ-4294: import type { SlackSettings } from "metabase-types/api";  
  
export const createMockSlackSettings = (  
 opts?: Partial<SlackSettings>,  
): SlackSettings => ({  
 "slack-app-token": null,  
 ...

REQ-4295: import type { NativeQuerySnippet } from "metabase-types/api";  
  
import { createMockEntityId } from "./entity-id";  
import { createMockUser } from "./user";  
  
export const createMockNativeQuerySnippet =...

REQ-4296: import type { StoreTokenStatus } from "metabase-types/api";  
  
export const createMockStoreTokenStatus = (  
 opts?: Partial<StoreTokenStatus>,  
): StoreTokenStatus => ({  
 ...opts,  
 valid: false,  
 ...

REQ-4297: import type { ForeignKey, Schema, Table } from "metabase-types/api";  
  
export const createMockTable = (opts?: Partial<Table>): Table => {  
 return {  
 id: 1,  
 db\_id: 1,  
 display\_name:...

REQ-4298: import type { Task } from "metabase-types/api";  
  
export const createMockTask = (task?: Partial<Task>): Task => ({  
 id: 1,  
 db\_id: 1,  
 duration: 100,  
 started\_at:...

REQ-4299: import type {  
 Timeline,  
 TimelineData,  
 TimelineEvent,  
 TimelineEventData,  
} from "../timeline";  
  
import { createMockUserInfo } from "./user";  
  
export const createMockTimeline = (opts?:...

REQ-4300: import type { Undo } from "metabase-types/store/undo";  
  
export const createMockUndo = (opts?: Partial<Undo>): Undo => ({  
 message: "The filter was auto-connected to all questions.",  
 actionLabel:...

REQ-4301: import type { User, UserInfo, UserListResult } from "metabase-types/api";  
  
export const createMockUser = (opts?: Partial<User>): User => ({  
 id: 1,  
 first\_name: "Testy",  
 last\_name: "Tableton",  
 ...

REQ-4302: import type { CardId } from "./card";  
import type { FieldReference } from "./query";  
  
export type NormalizedIndexedEntity = IndexedEntity;  
  
export type ModelIndex = {  
 id: number;  
 model\_id:...

REQ-4303: import type {  
 BaseUser,  
 CardId,  
 CollectionAuthorityLevel,  
 CollectionId,  
 DatabaseId,  
 UserId,  
} from "metabase-types/api";  
  
export type ModelCacheState =  
 | "creating"  
 | "refreshing"  
 |...

REQ-4304: import type { BaseUser } from "./user";  
  
export type VerifyItemRequest = {  
 status: "verified" | null;  
 moderated\_item\_id: number;  
 moderated\_item\_type: "card" | "dashboard";  
 text?:...

REQ-4305: import type {  
 ScheduleSettings,  
 ScheduleType,  
} from "metabase-types/api/settings";  
import type { User } from "metabase-types/api/user";  
import { isObject } from "metabase-types/guards";  
  
type...

REQ-4306: import type { Card, CardId } from "metabase-types/api/card";  
import type { Channel } from "metabase-types/api/notification-channels";  
import type { PaginationRequest } from...

REQ-4307: export interface PaginationRequest {  
 limit?: number | null;  
 offset?: number | null;  
}  
  
export interface PaginationResponse {  
 limit: number | null;  
 offset: number | null;  
 total: number;  
}

REQ-4308: import type { CardId } from "./card";  
import type { RowValue, TemporalUnit } from "./dataset";  
import type { FieldReference } from "./query";  
  
export type StringParameterType =  
 | "string/="  
 |...

REQ-4309: \* 'Model' as in 'type of object'

REQ-4310: import type {  
 DataPermission,  
 DataPermissionValue,  
} from "metabase/admin/permissions/types";  
import type {  
 CollectionId,  
 DatabaseId,  
 SchemaName,  
 TableId,  
} from...

REQ-4311: import type { ModelCacheRefreshStatus } from "./models";  
import type { PaginationRequest, PaginationResponse } from "./pagination";  
  
export type PersistedInfoId = number;  
  
export type...

REQ-4312: import type { Card } from "./card";  
import type { Channel } from "./notification-channels";  
  
export type Pulse = {  
 cards: Card[];  
 channels: Channel[];  
 name?: string;  
 parameters?:...

REQ-4313: import type { UiParameter } from "metabase-lib/v1/parameters/types";  
import type {  
 CardId,  
 DatabaseId,  
 FieldId,  
 SegmentId,  
 TableId,  
 TemplateTags,  
} from "metabase-types/api";  
  
export...

REQ-4314: import type { CardId } from "metabase-types/api";  
  
export interface Revision {  
 id: number;  
 description: string;  
 message: string | null;  
 timestamp: string;  
 is\_creation: boolean;  
 ...

REQ-4315: import type { WritebackAction } from "./actions";  
import type { Card, CardId } from "./card";  
import type { Collection, CollectionId, CollectionItemId } from "./collection";  
import type { Dashboard }...

REQ-4316: import type { UserId } from "metabase-types/api/user";  
  
import type { CardId } from "./card";  
import type { Collection, CollectionId, LastEditInfo } from "./collection";  
import type { Dashboard,...

REQ-4317: import type { StructuredQuery } from "./query";  
import type { Table, TableId } from "./table";  
  
export type SegmentId = number;  
  
export interface Segment {  
 id: SegmentId;  
 name: string;  
 ...

REQ-4318: export interface PasswordResetTokenStatus {  
 valid: boolean;  
}

REQ-4319: import type { ReactNode } from "react";  
  
import type { SdkIframeEmbedSetupSettings } from "metabase-enterprise/embedding\_iframe\_sdk\_setup/types";  
  
import type { InputSettingType } from...

REQ-4320: export type UsageReason =  
 | "self-service-analytics"  
 | "embedding"  
 | "both"  
 | "not-sure";

REQ-4321: export interface SlackSettings {  
 "slack-app-token": string | null;  
 "slack-bug-report-channel": string | null;  
}

REQ-4322: import type { RegularCollectionId } from "./collection";  
import type { BaseEntityId } from "./entity-id";  
import type { UserId, UserInfo } from "./user";  
  
export type NativeQuerySnippetId =...

REQ-4323: export enum SortDirection {  
 Asc = "asc",  
 Desc = "desc",  
}  
  
export type SortingOptions<SortColumn extends string> = {  
 sort\_column: SortColumn;  
 sort\_direction: SortDirection;  
};

REQ-4324: export interface StoreTokenStatus {  
 status?: string;  
 valid: boolean;  
 trial: boolean;  
}  
  
export const supportedFormatTypes = [  
 "string",  
 "integer",  
 "float",  
 "datetime",  
 "currency",  
]...

REQ-4325: import type { Card } from "./card";  
import type { RegularCollectionId } from "./collection";  
import type { DashboardId } from "./dashboard";  
import type { BaseEntityId } from "./entity-id";  
import...

REQ-4326: import type { Card, CardType } from "./card";  
import type { Database, DatabaseId, InitialSyncStatus } from "./database";  
import type { Field, FieldDimensionOption, FieldId } from "./field";  
import...

REQ-4327: import type { DatabaseId } from "./database";  
import type { PaginationRequest, PaginationResponse } from "./pagination";  
import type { SortingOptions } from "./sorting";  
  
// "unknown" status is only...

REQ-4328: import type { CardId } from "./card";  
import type {  
 Collection,  
 CollectionId,  
 RegularCollectionId,  
} from "./collection";  
import type { UserInfo } from "./user";  
  
export type TimelineId =...

REQ-4329: import type { CollectionId } from "./collection";  
import type { DashboardId } from "./dashboard";  
import type { PaginationRequest, PaginationResponse } from "./pagination";  
  
export type UserId =...

REQ-4330: export interface Log {  
 timestamp: string;  
 process\_uuid: string;  
 fqns: string;  
 msg: string;  
 level: string;  
 exception: any;  
}  
  
export interface MetabaseInfo {  
 "application-database":...

REQ-4331: SmartScalar (Trend Chart)

REQ-4332: export const virtualCardDisplayTypes = [  
 "action",  
 "heading",  
 "link",  
 "placeholder",  
 "text",  
 "iframe",  
] as const;  
  
export type VirtualCardDisplay = (typeof...

REQ-4333: import type {  
 VisualizationDisplay,  
 VisualizationSettings,  
} from "metabase-types/api";  
  
export type VisualizerDataSourceType = "card";  
export type VisualizerDataSourceId =...

REQ-4334: import type { IconProps } from "metabase/ui";  
import type { Collection } from "metabase-types/api";  
  
export type WrappedEntity<Entity> = {  
 getName: () => string;  
 getIcon: () => IconProps;  
 ...

REQ-4335: import type { Database } from "metabase-types/api";  
  
export interface DatabaseEntity extends Database {  
 fetchIdFields: (query: any) => void;  
}

REQ-4336: export \* from "./table";  
export \* from "./database";  
export \* from "./common";

REQ-4337: import type { Table } from "metabase-types/api";  
  
export interface TableEntity extends Table {  
 updateProperty: (name: string, value: string | number | null) => void;  
}

REQ-4338: \*

REQ-4339: import type { Card, UnsavedCard } from "metabase-types/api";  
  
export function isSavedCard(card: Card | UnsavedCard): card is Card {  
 return "id" in card && card.id != null;  
}

REQ-4340: import type {  
 BaseActionClickBehavior,  
 DeleteActionClickBehavior,  
 ImplicitActionClickBehavior,  
 InsertActionClickBehavior,  
 UpdateActionClickBehavior,  
} from "metabase-types/api";  
  
import {...

REQ-4341: export const isObject = (  
 value: unknown,  
): value is Record<string | number | symbol, unknown> => {  
 return typeof value === "object" && value !== null;  
};

REQ-4342: import type { DictionaryArray, DictionaryArrayRow } from "metabase-types/api";  
  
export const isDictionaryArrayRow = (  
 item: unknown,  
): item is DictionaryArrayRow => {  
 return (  
 typeof item...

REQ-4343: import type {  
 LinkEntity,  
 RestrictedLinkEntity,  
} from "metabase-types/api/dashboard";  
  
export const isRestrictedLinkEntity = (  
 value: LinkEntity,  
): value is RestrictedLinkEntity =>  
 !!(value...

REQ-4344: import type {  
 DateTimeAbsoluteUnit,  
 DateTimeRelativeUnit,  
} from "metabase-types/api";  
import {  
 dateTimeAbsoluteUnits,  
 dateTimeRelativeUnits,  
} from "metabase-types/api";  
  
export const...

REQ-4345: export function isElement(a: any): a is Element {  
 return a instanceof Element;  
}

REQ-4346: import type {  
 CustomFormFieldDefinition,  
 FormFieldDefinition,  
 StandardFormFieldDefinition,  
} from "metabase-types/forms";  
  
import { isReactComponent } from "./react";  
  
export function...

REQ-4347: export \* from "./card";  
export \* from "./click-behavior";  
export \* from "./common";  
export \* from "./dom";  
export \* from "./settings";  
export \* from "./forms";  
export \* from "./parameters";  
export \*...

REQ-4348: import type {  
 ParameterDimensionTarget,  
 ParameterTarget,  
 StructuredParameterDimensionTarget,  
} from "metabase-types/api";  
  
export function isDimensionTarget(  
 target: ParameterTarget |...

REQ-4349: import type \* as React from "react";  
import \* as ReactIs from "react-is";  
  
// checking to see if the `element` is in JSX.IntrinisicElements since they support refs  
// tippy's `children` prop seems to...

REQ-4350: Type Guards

REQ-4351: import { type EngineKey, engineKeys } from "metabase-types/api";  
  
export function isEngineKey(value: string | undefined): value is EngineKey {  
 return engineKeys.includes(value as EngineKey);  
}

REQ-4352: import type {  
 CollectionPermissions,  
 DatabaseId,  
 GroupsPermissions,  
 SettingDefinition,  
} from "metabase-types/api";  
  
export type AdminPathKey =  
 | "data-model"  
 | "settings"  
 | "metabot"  
 ...

REQ-4353: import type { ChecklistItemValue } from "metabase/home/components/Onboarding/types";  
import type { CollectionId } from "metabase-types/api/collection";  
  
export interface AppErrorDescriptor {  
 ...

REQ-4354: export interface AuthState {  
 loginPending: boolean;  
 redirect: boolean;  
}

REQ-4355: import type { IconName } from "metabase/ui";  
import type { Collection, CollectionId } from "metabase-types/api";  
  
// see entities/collections/getExpandedCollectionsById.js  
export type...

REQ-4356: import type { DisplayTheme } from "metabase/public/lib/types";  
import type {  
 DashCardDataMap,  
 DashCardId,  
 Dashboard,  
 DashboardCard,  
 DashboardId,  
 DashboardTab,  
 DashboardTabId,  
 ...

REQ-4357: export interface Download {  
 id: number;  
 title: string;  
 status: "complete" | "in-progress" | "error";  
 error?: string;  
}  
  
export type DownloadsState = Download[];

REQ-4358: import type {  
 EmbeddingDataPicker,  
 EmbeddingEntityType,  
} from "./embedding-data-picker";  
  
export interface InteractiveEmbeddingOptions {  
 font: string | undefined;  
 top\_nav: boolean;  
 search:...

REQ-4359: export interface EmbeddingDataPickerState {  
 entityTypes: EmbeddingEntityType[];  
}  
  
/\*\*  
 \* `question` only works on multi-stage data picker, not the simple data picker.  
 \* The reason being that we...

REQ-4360: import type {  
 NormalizedCard,  
 NormalizedCollection,  
 NormalizedDashboard,  
 NormalizedDatabase,  
 NormalizedField,  
 NormalizedIndexedEntity,  
 NormalizedNativeQuerySnippet,  
 NormalizedSchema,  
...

REQ-4361: export \* from "./admin";  
export \* from "./app";  
export \* from "./auth";  
export \* from "./collection";  
export \* from "./dashboard";  
export \* from "./embed";  
export \* from "./entities";  
export \* from...

REQ-4362: import type { AdminAppState, AdminState } from "metabase-types/store";  
  
export const createMockAdminState = (  
 opts?: Partial<AdminState>,  
): AdminState => ({  
 app: createMockAdminAppState(),  
 ...

REQ-4363: import type { AppState } from "metabase-types/store";  
  
export const createMockAppState = (opts?: Partial<AppState>): AppState => ({  
 isNavbarOpen: true,  
 errorPage: null,  
 isDndAvailable: false,  
 ...

REQ-4364: import type { AuthState } from "metabase-types/store";  
  
export const createMockAuthState = (opts?: Partial<AuthState>): AuthState => ({  
 loginPending: false,  
 redirect: true,  
 ...opts,  
});

REQ-4365: import { createMockDashboard } from "metabase-types/api/mocks";  
import type { DashboardState, StoreDashboard } from "metabase-types/store";  
  
export const createMockDashboardState = (  
 opts:...

REQ-4366: import type { Download } from "../downloads";  
  
export const createMockDownload = (props: Partial<Download> = {}): Download => {  
 return {  
 id: Date.now(),  
 title: "file.csv",  
 status:...

REQ-4367: import { DEFAULT\_INTERACTIVE\_EMBEDDING\_OPTIONS } from "metabase/redux/embed";  
import type {  
 EmbedState,  
 InteractiveEmbeddingOptions,  
} from "metabase-types/store";  
  
export const...

REQ-4368: import { DEFAULT\_EMBEDDING\_ENTITY\_TYPES } from "metabase/redux/embedding-data-picker";  
  
import type { EmbeddingDataPickerState } from "../embedding-data-picker";  
  
export const...

REQ-4369: import type { EntitiesState } from "metabase-types/store";  
  
// This is a helper for cases when entities state doesn't matter  
// Most likely, createMockEntitiesState from \_\_support\_\_/store would be a...

REQ-4370: export \* from "./admin";  
export \* from "./app";  
export \* from "./auth";  
export \* from "./dashboard";  
export \* from "./entities";  
export \* from "./embed";  
export \* from "./parameters";  
export \* from...

REQ-4371: import type { ParametersState } from "metabase-types/store/parameters";  
  
export const createMockParametersState = (  
 opts?: Partial<ParametersState>,  
): ParametersState => ({  
 parameterValuesCache:...

REQ-4372: import type {  
 QueryBuilderDashboardState,  
 QueryBuilderState,  
 QueryBuilderUIControls,  
} from "metabase-types/store";  
  
export const createMockQueryBuilderUIControlsState = (  
 opts?:...

REQ-4373: import type { RequestsState } from "../requests";  
  
export const createMockRequestsState = (): RequestsState => {  
 return {  
 entities: {},  
 plugins: {},  
 };  
};

REQ-4374: import type { Location } from "history";  
import type { RouterState } from "react-router-redux";  
  
export const createMockRoutingState = (  
 opts?: Partial<RouterState>,  
): RouterState => {  
 return {  
...

REQ-4375: import type { EnterpriseSettings, Settings } from "metabase-types/api";  
import { createMockSettings } from "metabase-types/api/mocks";  
import type { SettingsState } from...

REQ-4376: import type {  
 InviteInfo,  
 Locale,  
 SetupState,  
 SubscribeInfo,  
 UserInfo,  
} from "metabase-types/store";  
  
export const createMockLocale = (opts?: Partial<Locale>): Locale => ({  
 name:...

REQ-4377: import type { SdkStoreState } from "embedding-sdk/store/types";  
import type { EnterpriseState } from "metabase-enterprise/settings/types";  
import { createMockUser } from...

REQ-4378: import type { FileUpload } from "../upload";  
  
export const createMockUploadState = (uploads = {}) => {  
 return { ...uploads };  
};  
  
export const createMockUpload = (props?: Partial<FileUpload>):...

REQ-4379: import type { VisualizerState } from "../visualizer";  
  
export const createMockVisualizerState = (  
 opts?: Partial<VisualizerState>,  
): VisualizerState => ({  
 initialState: {  
 display: null,  
 ...

REQ-4380: import type { ParameterValues } from "metabase-types/api";  
  
export type ParameterValuesCache = Record<string, ParameterValues>;  
  
export interface ParametersState {  
 parameterValuesCache:...

REQ-4381: import type { Deferred } from "metabase/lib/promise";  
import type { QueryModalType } from "metabase/query\_builder/constants";  
import type { Widget } from...

REQ-4382: type EntityKey = string;  
  
type QueryKey = string;  
  
type RequestType = string; // only "fetch"?  
  
// See initialRequestState in metabase/redux/requests.js  
export interface RequestState {  
 loading:...

REQ-4383: import type { EnterpriseSettings } from "metabase-types/api";  
  
export interface SettingsState {  
 values: EnterpriseSettings;  
 loading: boolean;  
}

REQ-4384: import type { SetupStep } from "metabase/setup/types";  
import type { DatabaseData, UsageReason } from "metabase-types/api";  
  
export interface Locale {  
 name: string;  
 code: string;  
}  
  
export...

REQ-4385: import type { RouterState } from "react-router-redux";  
  
import type { User } from "metabase-types/api";  
  
import type { AdminState } from "./admin";  
import type { AppState } from "./app";  
import type...

REQ-4386: import type { ReactNode, RefObject } from "react";  
  
import type { IconName } from "metabase/ui";  
import type { DashCardId, DashboardTabId } from "metabase-types/api";  
  
export interface Undo {  
 id:...

REQ-4387: import type { CollectionId, TableId } from "metabase-types/api";  
  
export enum UploadMode {  
 append = "append",  
 create = "create",  
 replace = "replace",  
}  
  
export type FileUpload = {  
 status:...

REQ-4388: import type {  
 Card,  
 Dataset,  
 DatasetColumn,  
 VisualizerDataSource,  
 VisualizerDataSourceId,  
 VisualizerVizDefinition,  
} from "metabase-types/api";  
  
type BaseDraggedItem<T> = {  
 id: string;  
...

REQ-4389: import type { ManipulateType } from "dayjs";  
  
type SupportedUnit = QUnitType | "isoWeek" | ManipulateType;  
  
declare module "dayjs" {  
 interface Dayjs {  
 add(value: number, unit: SupportedUnit):...

REQ-4390: import "@emotion/react";  
  
import type { MantineTheme } from "@mantine/core";  
interface \_EmotionCompatibilityTheme {  
 fn: { themeColor: (colorName: string) => string };  
}  
  
declare module...

REQ-4391: interface Window {  
 MetabaseBootstrap: any;  
 MetabaseRoot?: string;  
 MetabaseNonce?: string;  
}  
  
// This allows importing static SVGs from TypeScript files  
declare module "\*.svg" {  
 const content:...

REQ-4392: declare module "\*.svg" {  
 const value: string;  
  
 // eslint-disable-next-line import/no-default-export  
 export default value;  
}  
  
declare module "\*.svg?source" {  
 const value: string;  
  
 //...

REQ-4393: import type { EmbeddingThemeOptions } from "metabase/embedding-sdk/theme/private";  
  
interface \_EmotionCompatibilityTheme {  
 fn: {  
 themeColor: (colorName: string) => string;  
 };  
}  
  
declare...

REQ-4394: declare module "react-ansi-style";

REQ-4395: import type \* as React from "react";  
  
declare module "react" {  
 // to make forwardRef work with generic function components  
 // eslint-disable-next-line @typescript-eslint/ban-types  
 function...

REQ-4396: declare module "slugg" {  
 type Separator = string;  
 type ToStrip = string | RegExp;  
  
 type Options = {  
 separator?: Separator;  
 toStrip?: ToStrip;  
 toLowerCase?: boolean;  
 };  
  
 //...

REQ-4397: \*

REQ-4398: {  
 "rules": {  
 "import/no-commonjs": 0,  
 "no-color-literals": 0  
 },  
 "env": {  
 "node": true  
 }  
}

REQ-4399: module.exports = {};

REQ-4400: module.exports = "test-file-stub";

REQ-4401: module.exports = {};

REQ-4402: !/usr/bin/env bash

REQ-4403: import userEvent from "@testing-library/user-event";  
  
import { screen, within } from "\_\_support\_\_/ui";  
  
export type ViewMantineSelectOptionsParams = {  
 /\*\* This function will identify the root...

REQ-4404: import \* as EnterpriseContentTranslationUtilsModule from "metabase-enterprise/content\_translation/utils";  
  
/\*\*  
 \* One of the utility functions that makes the content translation feature tick  
 \* is...

REQ-4405: import type {  
 BreakoutSeriesModel,  
 SeriesModel,  
} from "metabase/visualizations/echarts/cartesian/model/types";  
import { createMockColumn } from "metabase-types/api/mocks";  
  
export const...

REQ-4406: \*

REQ-4407: import { combineReducers, configureStore } from "@reduxjs/toolkit";  
import promise from "redux-promise";  
  
import \* as entities from "metabase/redux/entities";  
import requestsReducer from...

REQ-4408: type MockEvent = {  
 preventDefault: jest.Mock;  
 returnValue?: string;  
};  
  
type CallMockEventType = (  
 mockEventListener: jest.SpyInstance,  
 eventName: string,  
) => MockEvent;  
  
// calls event...

REQ-4409: import { getMetadata } from "metabase/selectors/metadata";  
import type { Settings } from "metabase-types/api";  
import {  
 createMockSettingsState,  
 createMockState,  
} from...

REQ-4410: global.ga = () => {};  
global.snowplow = () => {};  
  
global.window.matchMedia = () => ({  
 addEventListener: () => {},  
 removeEventListener: () => {},  
});  
  
/\*\*  
 \* jsdom doesn't have scrollBy or...

REQ-4411: import fetchMock from "fetch-mock";  
  
import type {  
 CardId,  
 GetPublicAction,  
 WritebackAction,  
} from "metabase-types/api";  
import {  
 createMockImplicitQueryAction,  
 createMockQueryAction,  
}...

REQ-4412: import fetchMock, { type UserRouteConfig } from "fetch-mock";  
  
import type {  
 Dashboard,  
 PopularItem,  
 RecentContexts,  
 RecentItem,  
} from "metabase-types/api";  
  
export function...

REQ-4413: import fetchMock from "fetch-mock";  
  
import type { AIEntityAnalysisResponse } from "metabase-types/api";  
  
export function setupAnalyzeChartEndpoint(response: AIEntityAnalysisResponse) {  
 const name...

REQ-4414: import fetchMock from "fetch-mock";  
  
import type { FixSqlQueryResponse } from "metabase-types/api";  
  
export function setupFixNativeQueryEndpoint(response: FixSqlQueryResponse) {  
 ...

REQ-4415: import fetchMock from "fetch-mock";  
  
import type { GenerateSqlQueryResponse } from "metabase-types/api";  
  
export function setupGenerateSqlQueryEndpoint(  
 response: GenerateSqlQueryResponse,  
) {  
 ...

REQ-4416: import fetchMock from "fetch-mock";  
  
import type { Alert, Card } from "metabase-types/api";  
  
export function setupAlertsEndpoints(card: Card, alerts: Alert[]) {  
 ...

REQ-4417: import fetchMock from "fetch-mock";  
  
import type { ApiKey } from "metabase-types/api";  
  
export function setupApiKeyEndpoints(apiKeys: ApiKey[]) {  
 fetchMock.get("path:/api/api-key", apiKeys);  
 ...

REQ-4418: import fetchMock, { type UserRouteConfig } from "fetch-mock";  
  
import type {  
 CardId,  
 CollectionId,  
 DashboardId,  
 User,  
} from "metabase-types/api";  
  
interface AuditInfo {  
 dashboard\_overview:...

REQ-4419: import fetchMock from "fetch-mock";  
  
import type { DatabaseId, DatabaseXray } from "metabase-types/api";  
  
export function setupDatabaseCandidatesEndpoint(  
 id: DatabaseId,  
 candidates:...

REQ-4420: import fetchMock from "fetch-mock";  
  
import type { Bookmark } from "metabase-types/api";  
  
export function setupBookmarksEndpoints(bookmarks: Bookmark[]) {  
 fetchMock.get("path:/api/bookmark",...

REQ-4421: import fetchMock from "fetch-mock";  
  
export const setupBugReportEndpoints = (responses = [{ success: true }]) => {  
 fetchMock.post(  
 "path:/api/slack/bug-report",  
 { status: 200, body:...

REQ-4422: import fetchMock from "fetch-mock";  
  
import { getQuestionVirtualTableId } from "metabase-lib/v1/metadata/utils/saved-questions";  
import type {  
 Card,  
 CardId,  
 CardQueryMetadata,  
 Dataset,  
 ...

REQ-4423: import fetchMock from "fetch-mock";  
  
import type { NotificationChannel } from "metabase-types/api";  
  
export const setupWebhookChannelsEndpoint = (  
 channels: NotificationChannel[] = [],  
) => {  
 ...

REQ-4424: import fetchMock from "fetch-mock";  
import \_ from "underscore";  
  
import { ROOT\_COLLECTION } from "metabase/entities/collections/constants";  
import {  
 SAVED\_QUESTIONS\_VIRTUAL\_DB\_ID,  
 ...

REQ-4425: export const PERMISSION\_ERROR = "You don't have permissions to do that.";

REQ-4426: import fetchMock from "fetch-mock";  
  
import type { DictionaryArray, DictionaryResponse } from "metabase-types/api";  
  
export function setupContentTranslationEndpoints({  
 dictionary = [],  
 ...

REQ-4427: import fetchMock from "fetch-mock";  
  
import { getNextId } from "\_\_support\_\_/utils";  
import type {  
 Dashboard,  
 DashboardId,  
 DashboardQueryMetadata,  
 FieldId,  
 GetPublicDashboard,  
} from...

REQ-4428: import fetchMock from "fetch-mock";  
  
import type { DashboardId, Dataset } from "metabase-types/api";  
import type { StoreDashcard } from "metabase-types/store";  
  
export function...

REQ-4429: import fetchMock from "fetch-mock";  
import \_ from "underscore";  
  
import { SAVED\_QUESTIONS\_DATABASE } from "metabase/databases/constants";  
import { isTypePK } from...

REQ-4430: import fetchMock from "fetch-mock";  
  
import type { CardQueryMetadata, ParameterValues } from "metabase-types/api";  
import type { MockDatasetOpts } from "metabase-types/api/mocks";  
import {...

REQ-4431: email settings, why aren't they in the settings endpoint? who knows? \\_()\_/

REQ-4432: import fetchMock from "fetch-mock";  
  
import type { Card, Dashboard, DashboardCard } from "metabase-types/api";  
import { createMockDataset } from "metabase-types/api/mocks";  
  
export function...

REQ-4433: import fetchMock from "fetch-mock";  
  
import type { TranslateEntityIdRequest } from "metabase/api";  
import type {  
 BaseEntityId,  
 CardId,  
 CollectionId,  
 DashboardId,  
} from...

REQ-4434: import fetchMock from "fetch-mock";  
  
import type {  
 Field,  
 FieldId,  
 FieldValue,  
 GetFieldValuesResponse,  
} from "metabase-types/api";  
import { createMockFieldValues } from...

REQ-4435: import fetchMock from "fetch-mock";  
  
import type { GdrivePayload } from "metabase-types/api";  
  
export function setupGdriveGetFolderEndpoint({  
 errorCode,  
 ...gdrivePayload  
}: Partial<GdrivePayload>...

REQ-4436: import fetchMock from "fetch-mock";  
import type { FeatureCollection } from "geojson";  
  
export function setupGeoJSONEndpoint({  
 featureCollection,  
 url,  
}: {  
 featureCollection: FeatureCollection;  
...

REQ-4437: import fetchMock from "fetch-mock";  
  
export function setupUpdateGoogleAuthEndpoint(  
 { status }: { status?: number } = { status: 204 },  
) {  
 fetchMock.put(  
 new RegExp("/api/google/settings"),  
 ...

REQ-4438: import fetchMock from "fetch-mock";  
  
import type { Group } from "metabase-types/api";  
  
export const setupGroupsEndpoint = (groups: Omit<Group, "members">[]) => {  
 ...

REQ-4439: import fetchMock from "fetch-mock";  
  
import type { DatabaseId, GroupId, Impersonation } from "metabase-types/api";  
  
export const setupExistingImpersonationEndpoint = (  
 impersonation:...

REQ-4440: export \* from "./action";  
export \* from "./activity";  
export \* from "./ai-entity-analysis";  
export \* from "./ai-sql-fixer";  
export \* from "./ai-sql-generation";  
export \* from "./alert";  
export \* from...

REQ-4441: import fetchMock from "fetch-mock";  
  
import type { LoggerPreset } from "metabase-types/api";  
  
export function setupLoggerPresetsEndpoint(response: LoggerPreset[]) {  
 ...

REQ-4442: import fetchMock, { type UserRouteConfig } from "fetch-mock";  
  
import type {  
 MetabotApiEntity,  
 MetabotId,  
 MetabotInfo,  
 SuggestedMetabotPrompt,  
 SuggestedMetabotPromptsResponse,  
} from...

REQ-4443: import fetchMock from "fetch-mock";  
  
import type { CardId, ModelIndex } from "metabase-types/api";  
import { createMockModelIndex } from "metabase-types/api/mocks";  
  
export function...

REQ-4444: import fetchMock from "fetch-mock";  
  
import type { NativeQuerySnippet } from "metabase-types/api";  
  
const PATH = "path:/api/native-query-snippet";  
  
export function setupNativeQuerySnippetEndpoints(  
 ...

REQ-4445: import fetchMock from "fetch-mock";  
  
import type {  
 ListNotificationsRequest,  
 Notification,  
} from "metabase-types/api";  
  
export const setupListNotificationEndpoints = (  
 { card\_id }:...

REQ-4446: import fetchMock from "fetch-mock";  
  
import type { CacheConfig } from "metabase-types/api";  
export function setupPerformanceEndpoints(cacheConfigs: CacheConfig[]) {  
 fetchMock.get("path:/api/cache",...

REQ-4447: import fetchMock from "fetch-mock";  
  
import type {  
 CollectionPermissionsGraph,  
 Database,  
 Group,  
} from "metabase-types/api";  
import { createMockPermissionsGraph } from...

REQ-4448: import fetchMock from "fetch-mock";  
  
import type { ModelCacheRefreshStatus } from "metabase-types/api";  
  
export function setupModelPersistenceEndpoints(  
 persistedModels:...

REQ-4449: import fetchMock from "fetch-mock";  
  
export const setupTokenStatusEndpoint = ({  
 valid,  
 features,  
}: {  
 valid: boolean;  
 features?: string[];  
}) => {  
 const name = "premium-token-status";  
 ...

REQ-4450: import fetchMock from "fetch-mock";  
  
import type { EmbedDataset, PublicCard } from "metabase-types/api";  
  
export function setupPublicQuestionEndpoints(  
 uuid: string,  
 publicCard: PublicCard,  
) {  
 ...

REQ-4451: import fetchMock from "fetch-mock";  
  
import type { ChannelApiResponse } from "metabase-types/api";  
  
export const setupNotificationChannelsEndpoints = (  
 channelData:...

REQ-4452: import fetchMock from "fetch-mock";  
  
import type { Revision } from "metabase-types/api";  
  
export function setupRevisionsEndpoints(revisions: Revision[]) {  
 fetchMock.get("path:/api/revision",...

REQ-4453: import fetchMock from "fetch-mock";  
import { match } from "ts-pattern";  
  
import type { CollectionItem, SearchResult } from "metabase-types/api";  
import type { EmbeddingDataPicker } from...

REQ-4454: import fetchMock from "fetch-mock";  
  
import type { Segment } from "metabase-types/api";  
import { createMockSegment } from "metabase-types/api/mocks";  
  
export function setupSegmentEndpoint(segment:...

REQ-4455: import fetchMock from "fetch-mock";  
  
import type {  
 EnterpriseSettings,  
 PasswordResetTokenStatus,  
 Settings,  
} from "metabase-types/api";  
  
export function setupPropertiesEndpoints(  
 settings:...

REQ-4456: import fetchMock from "fetch-mock";  
  
import type {  
 MaskedScimApiKey,  
 UnmaskedScimApiKey,  
} from "metabase-enterprise/user\_provisioning/types";  
import type {  
 EnterpriseSettingKey,  
 ...

REQ-4457: import fetchMock from "fetch-mock";  
  
export function setupErrorSetupEndpoints() {  
 fetchMock.post("path:/api/setup", 400);  
}  
  
export function setupForTokenCheckEndpoint(response: { valid: boolean })...

REQ-4458: import fetchMock from "fetch-mock";  
  
export function setupSlackManifestEndpoint() {  
 fetchMock.get("path:/api/slack/manifest", "manifest-content");  
}  
  
export function setupSlackSettingsEndpoint() {  
...

REQ-4459: import fetchMock from "fetch-mock";  
  
import type { StoreTokenStatus } from "metabase-types/api";  
  
export function setupStoreTokenEndpoints(tokenStatus: StoreTokenStatus) {  
 ...

REQ-4460: import fetchMock from "fetch-mock";  
  
import type { ForeignKey, Table } from "metabase-types/api";  
  
import { setupFieldEndpoints } from "./field";  
  
export function setupTableEndpoints(  
 table:...

REQ-4461: import fetchMock, { type UserRouteConfig } from "fetch-mock";  
  
import type { ListTasksResponse, Task } from "metabase-types/api";  
  
export function setupTasksEndpoints(  
 response: ListTasksResponse,  
...

REQ-4462: import fetchMock from "fetch-mock";  
  
import type { Timeline } from "metabase-types/api";  
  
export function setupTimelinesEndpoints(timelines: Timeline[]) {  
 fetchMock.get("path:/api/timeline",...

REQ-4463: import fetchMock from "fetch-mock";  
  
import type { UserKeyValue, UserKeyValueKey } from "metabase-types/api";  
  
export const setupUserKeyValueEndpoints = ({  
 namespace,  
 key,  
 value,  
}:...

REQ-4464: import fetchMock from "fetch-mock";  
  
import type {  
 User,  
 UserAttributeKey,  
 UserListResult,  
} from "metabase-types/api";  
  
export function setupUserEndpoints(user: UserListResult) {  
 ...

REQ-4465: import fetchMock from "fetch-mock";  
  
export function setupPasswordCheckEndpoint() {  
 fetchMock.post("path:/api/session/password-check", 204);  
}  
  
type ResponseInfo = {  
 url: string;  
 body:...

REQ-4466: import MetabaseSettings from "metabase/lib/settings";  
import type { EnterpriseSettings, Settings } from "metabase-types/api";  
import { createMockSettings } from "metabase-types/api/mocks";  
import {...

REQ-4467: import type { Schema as NormalizrSchema } from "normalizr";  
import { normalize } from "normalizr";  
  
import {  
 ActionSchema,  
 CollectionSchema,  
 DashboardSchema,  
 DatabaseSchema,  
 FieldSchema,  
 ...

REQ-4468: import type { DatasetColumn, DatasetData } from "metabase-types/api";  
import { createMockDatasetData } from "metabase-types/api/mocks";  
import { PRODUCTS, PRODUCTS\_ID } from...

REQ-4469: export default function testAcrossTimezones(runTests) {  
 // run\_timezone\_tests sets "TZ" environment variable to change the timezone  
 const clientTz = process.env["TZ"] || "[default]";  
 //...

REQ-4470: jest.mock("metabase/common/components/Popover");  
  
// Replace addEventListener with a test implementation which collects all event listeners to `eventListeners` map  
export const eventListeners =...

REQ-4471: import type { CallLog } from "fetch-mock";  
  
import { act, waitFor } from "./ui";  
  
export const getNextId = (() => {  
 let id = 0;  
 return (startingId?: number) => {  
 if (startingId) {  
 id =...

REQ-4472: import { getComputedSettingsForSeries } from "metabase/visualizations/lib/settings/visualization";  
import { formatValueForTooltip } from "metabase/visualizations/lib/tooltip";  
  
export function...

REQ-4473: const { readdir, writeFile } = require("fs");  
const { join: joinPath, relative } = require("path");  
  
const { promisify } = require("util");  
  
const asyncReaddir = promisify(readdir);  
const...

REQ-4474: import "@testing-library/jest-dom";  
import fetchMock from "fetch-mock";  
  
beforeEach(() => {  
 fetchMock.mockGlobal();  
});  
  
afterEach(() => {  
 fetchMock.removeRoutes();  
 ...

REQ-4475: eslint-disable no-undef

REQ-4476: import fetchMock from "fetch-mock";  
  
import api, { GET, POST, PUT } from "metabase/lib/api";  
api.basename = "";  
  
describe("api", () => {  
 const successResponse = { body: { status: "ok" } };  
  
 ...

REQ-4477: import { parseHashOptions, stringifyHashOptions } from "metabase/lib/browser";  
  
describe("browser", () => {  
 describe("parseHashOptions", () => {  
 it("should parse with prepended '#'", () => {  
 ...

REQ-4478: import {  
 createCard,  
 deserializeCardFromUrl,  
 serializeCardForUrl,  
} from "metabase/lib/card";  
import {  
 b64\_to\_utf8,  
 b64url\_to\_utf8,  
 utf8\_to\_b64,  
 utf8\_to\_b64url,  
} from...

REQ-4479: import { getPositionForNewDashCard } from "metabase/lib/dashboard\_grid";  
  
const getPos = (cards) => getPositionForNewDashCard(cards, 2, 2, 6);  
  
describe("dashboard\_grid", () => {  
 ...

REQ-4480: import \_ from "underscore";  
  
import {  
 COLLAPSED\_ROWS\_SETTING,  
 COLUMN\_SHOW\_TOTALS,  
 COLUMN\_SORT\_ORDER,  
 COLUMN\_SPLIT\_SETTING,  
 multiLevelPivot,  
 pivot,  
} from "metabase/lib/data\_grid";  
import...

REQ-4481: import {  
 getSelectionPosition,  
 parseDataUri,  
 setSelectionPosition,  
} from "metabase/lib/dom";  
  
describe("getSelectionPosition/setSelectionPosition", () => {  
 let container;  
  
 beforeEach(() =>...

REQ-4482: import moment from "moment-timezone"; // eslint-disable-line no-restricted-imports -- deprecated usage  
import { isElementOfType } from "react-dom/test-utils";  
  
import { mockSettings } from...

REQ-4483: import {  
 getActivePulseParameters,  
 getPulseParameters,  
 recipientIsValid,  
} from "metabase/lib/pulse";  
import MetabaseSettings from "metabase/lib/settings";  
  
describe("recipientIsValid", () =>...

REQ-4484: import {  
 foreignKeyCountsByOriginTable,  
 getSemanticTypeIcon,  
 getSemanticTypeName,  
} from "metabase/lib/schema\_metadata";  
import { TYPE } from...

REQ-4485: import { regexpEscape } from "metabase/lib/string";  
  
describe("regexpEscape", () => {  
 const testCases = [  
 ["nothing special here", "nothing special here"],  
 ["somewhat ./\\ special",...

REQ-4486: import moment from "moment-timezone"; // eslint-disable-line no-restricted-imports -- deprecated usage  
  
import { parseTime, parseTimestamp } from "metabase/lib/time";  
  
describe("time", () => {  
 ...

REQ-4487: import {  
 bookmark,  
 browseDatabase,  
 collection,  
 dashboard,  
 extractCollectionId,  
 extractEntityId,  
 extractQueryParams,  
 isCollectionPath,  
 model,  
 modelDetail,  
 modelEditor,  
 ...

REQ-4488: import { getFullName } from "metabase/lib/user";  
import { createMockUser } from "metabase-types/api/mocks";  
  
describe("lib/user", () => {  
 describe("getFullName", () => {  
 test("has both...

REQ-4489: import validate, { validators } from "metabase/lib/validate";  
  
describe("validators", () => {  
 describe("required", () => {  
 it("should return an error if the value is null", () => {  
 ...

REQ-4490: import "regenerator-runtime/runtime";  
import "metabase/css/vendor.css";  
import "metabase/css/index.module.css";  
  
window.MetabaseBootstrap = {  
 "enable-xrays": true,  
 "available-timezones": [  
 ...

REQ-4491: import { assoc, assocIn, dissoc } from "icepick";  
import { parse } from "url";  
  
import { createMockMetadata } from "\_\_support\_\_/metadata";  
import { deserializeCardFromUrl } from...

REQ-4492: import { createMockMetadata } from "\_\_support\_\_/metadata";  
import Database from "metabase-lib/v1/metadata/Database";  
import Table from "metabase-lib/v1/metadata/Table";  
import {  
 ORDERS\_ID,  
 ...

REQ-4493: import { normalize } from "cljs/metabase.legacy\_mbql.js";  
  
// This test is here mostly to make sure the shared CLJS lib works correctly.  
  
describe("normalize", () => {  
 it("should normalize an MBQL...

REQ-4494: import { assocIn } from "icepick";  
  
import { createMockMetadata } from "\_\_support\_\_/metadata";  
import Question from "metabase-lib/v1/Question";  
import NativeQuery, {  
 updateCardTemplateTagNames,  
}...

REQ-4495: import { memoizeClass, sortObject } from "metabase-lib/v1/utils";  
  
describe("sortObject", () => {  
 it("should serialize identically regardless of property creation order", () => {  
 const o1 =...

REQ-4496: We need to mock this \*before\* registering the visualizations. Otherwise  
`ChartWithLegend` with already load the real one.

REQ-4497: @ts-check

REQ-4498: module.exports = {  
 diffingEngine: "looks-same",  
 storiesFilter: [  
 "DataGrid",  
 "static-viz",  
 "viz",  
 "^visualizations/shared",  
 "^app/embed",  
 "^design system",  
 ...

REQ-4500: FROM clickhouse/clickhouse-server:25.2-alpine  
COPY .docker/clickhouse/single\_node\_tls/certificates /etc/clickhouse-server/certs  
RUN chown clickhouse:clickhouse -R /etc/clickhouse-server/certs \  
 ...

REQ-4501: This directory (modules/drivers/clickhouse/) includes code from the  
Clickhouse Metabase Driver (https://github.com/ClickHouse/metabase-clickhouse-driver)  
that is licensed under the Apache License,...

REQ-4502: ClickHouse Driver for Metabae  
## Attribution

REQ-4503: This directory (modules/drivers/starburst/) includes code from the  
Starburst Metabase Driver (https://github.com/starburstdata/metabase-driver)  
that is licensed under the Apache License, Version...

REQ-4504: eslint-env node  
eslint-disable import/no-commonjs

REQ-4505: To release locally, copy this file to a .env and fill in all values

REQ-4506: {  
 "overrides": [  
 {  
 "files": ["\*.ts", "\*.js"],  
 "rules": {  
 "no-console": "off"  
 }  
 }  
 ]  
}

REQ-4507: node\_modules/  
.env  
cache.json  
\*.log  
dist/  
version.properties  
version-info.json  
v\*.html  
channels.html

REQ-4508: Metabase Release

REQ-4509: changelog preview only, doesn't publish anything

REQ-4510: \*

REQ-4511: import "dotenv/config";  
import type { Octokit } from "@octokit/rest";  
import dayjs from 'dayjs';  
  
import { sendBackportReminder } from "./slack";  
import type { Issue } from "./types";  
  
const...

REQ-4512: export const nonUserFacingLabels = [  
 '.CI & Tests',  
 '.Building & Releasing',  
];  
  
export const hiddenLabels = [  
 'Type:Documentation',  
];

REQ-4513: import type { GithubCheck, GithubProps, Issue, ReleaseProps } from "./types";  
import {  
 getLastReleaseTag,  
 getMilestoneName,  
 getNextVersions,  
 getOSSVersion,  
} from "./version-helpers";  
  
export...

REQ-4514: export \* from "./backports";  
export \* from "./github";  
export \* from "./linked-issues";  
export \* from "./milestones";  
export \* from "./release-notes";  
export \* from "./release-status";  
export \* from...

REQ-4515: export function getLinkedIssues(body: string) {  
 const matches = body.match(  
 /(close(s|d)?|fixe?(s|d)?|resolve(s|d)?)(:?) (#|https?:\/\/github\.com\/.+metabase\/issues\/)(\d+)/gi,  
 );  
  
 if...

REQ-4516: import { getBackportSourcePRNumber, getLinkedIssues, getPRsFromCommitMessage } from "./linked-issues";  
  
const closingKeywords = [  
 "Close",  
 "Closes",  
 "Closed",  
 "Fix",  
 "Fixes",  
 "Fixed",  
 ...

REQ-4517: import fs from "fs";  
  
import { graphql } from "@octokit/graphql";  
import \_ from "underscore";  
  
import { hiddenLabels, nonUserFacingLabels } from "./constants";  
import {  
 findMilestone,  
 ...

REQ-4518: import { getNextMilestone } from "./milestones";  
import type { Milestone } from "./types";  
  
describe('milestones', () => {  
 const testMilestones = [  
 { number: 577, title: '0.57.7' },  
 {...

REQ-4519: import {$} from 'zx';  
  
export async function getReleaseBranchCommits(  
 { versionNumber }: { versionNumber: number }  
) {  
 const lastMajorVersion = versionNumber - 1;  
  
 // find where the last branch...

REQ-4520: import fs from 'fs';  
  
import { $ } from 'zx';  
  
const releaseChannels = [  
 "nightly",  
 "beta",  
 "latest",  
] as const;  
  
const editions = ["oss", "ee"] as const;  
  
type CommitInfo = {  
 version:...

REQ-4521: import { generateReleaseLog } from './release-log';  
  
generateReleaseLog();

REQ-4522: import fs from 'fs';  
  
import { $ } from 'zx';  
  
import { issueNumberRegex } from './linked-issues';  
  
type CommitInfo = {  
 versions: string[],  
 message: string,  
 hash: string,  
 date:...

REQ-4523: import { processCommit } from './release-log';  
  
describe('Release Log', () => {  
 describe('processCommit', () => {  
 it('should return an array of commit info objects with versions', () => {  
 ...

REQ-4524: export const githubReleaseTemplate = `## Upgrading  
  
> Before you upgrade, back up your Metabase application database!  
  
Check out our [upgrading...

REQ-4525: import { match } from "ts-pattern";  
  
import { hiddenLabels, nonUserFacingLabels } from "./constants";  
import { getMilestoneIssues, hasBeenReleased } from "./github";  
import { issueNumberRegex } from...

REQ-4526: import {  
 categorizeIssues,  
 generateReleaseNotes,  
 getChangelogUrl,  
 getReleaseTitle,  
 getWebsiteChangelog,  
 markdownIssueLinks,  
} from "./release-notes";  
import {  
 githubReleaseTemplate,  
 ...

REQ-4527: eslint-disable no-console

REQ-4528: import 'dotenv/config';  
import fs from 'fs';  
  
import { WebClient } from '@slack/web-api';  
import dayjs from 'dayjs';  
import relativeTime from 'dayjs/plugin/relativeTime';  
import fetch from...

REQ-4529: import type { Octokit } from "@octokit/rest";  
  
export interface CacheType {  
 version?: string;  
 majorVersion?: string;  
 releaseBranch?: string;  
 commitHash?: string;  
 versionType?: "major" |...

REQ-4530: import type { GithubProps, Tag } from "./types";  
  
// https://regexr.com/7l1ip  
export const isValidVersionString = (versionString: string) => {  
 return...

REQ-4531: import type { Tag } from "./types";  
import {  
 filterOutNonSupportedPrereleaseIdentifier,  
 findNextPatchVersion,  
 getBuildRequirements,  
 getDotXs,  
 getEnterpriseVersion,  
 ...

REQ-4532: import fetch from "node-fetch";  
  
import { getMilestoneIssues } from "./github";  
import type {  
 Issue,  
 ReleaseChannel,  
 ReleaseProps,  
 VersionInfo,  
 VersionInfoFile,  
} from "./types";  
import {  
 ...

REQ-4533: import type { Issue, VersionInfoFile } from "./types";  
import { generateVersionInfoJson, getVersionInfoUrl, updateVersionInfoChannelJson, updateVersionInfoLatestJson } from...

REQ-4534: \*\* DO NOT DELETE \*\*  
  
This file works as a sentinel to allow us to check if we are running in a jar file on JVMs >= 24  
  
\*\* DO NOT DELETE \*\*

REQ-4535: Directory Contents  
# `rds-combined-ca-bundle.pem`

REQ-4536: 123456  
password  
12345678  
qwerty  
123456789  
12345  
1234  
111111  
1234567  
dragon  
123123  
baseball  
abc123  
football  
monkey  
letmein  
shadow  
master  
696969  
michael  
mustang  
666666  
qwertyuiop  
123321  
1234567890  
pussy...

REQ-4537: Copyright 2016 The Merriweather Project Authors (https://github.com/EbenSorkin/Merriweather), with Reserved Font Name "Merriweather".  
  
This Font Software is licensed under the SIL Open Font License,...

REQ-4538: ! iFrame Resizer (iframeSizer.min.js ) - v4.3.2 - 2021-04-26

REQ-4539: (function () {  
 window.Metabase = window.Metabase || {};  
 window.Metabase.AssetErrorLoad = function (tag) {  
 console.error(  
 `Could not download asset ${tag.src} from your metabase...

REQ-4540: (function() {  
 window.MetabaseBootstrap = JSON.parse(document.getElementById("\_metabaseBootstrap").textContent);  
 window.MetabaseUserLocalization =...

REQ-4541: const content = [  
 ["no-sql-required", "Make decisions with data - no SQL required"],  
 ["bring-data-to-slack", "Bring your charts and data into Slack"],  
 ["click-charts-dive-deeper", "Click on...

REQ-4542: var shared;!function(){var t={168:function(t,e,r){let n=r(874),a={};for(let t of Object.keys(n))a[n[t]]=t;let...

REQ-4543: const toJSArray = (a) => {  
 const jsArray = [];  
 for (let i = 0; i < a.length; i++) {  
 jsArray[i] = a[i];  
 }  
 return jsArray;  
};  
  
function toJSMap(m) {  
 const o = {};  
 for (let i = 0; i <...

REQ-4544: - \*\*The API is subject to change.\*\* We rarely change API endpoints, and almost never remove them, but if you write code that relies on the API, there's a chance you might have to update your code in...

REQ-4545: @ts-check  
eslint-env node  
eslint-disable import/no-commonjs

REQ-4546: eslint-env node  
eslint-disable import/no-commonjs  
eslint-disable import/order

REQ-4547: @ts-check  
eslint-disable no-undef

REQ-4548: eslint-env node  
eslint-disable import/no-commonjs  
eslint-disable import/order

REQ-4549: eslint-disable import/no-commonjs  
eslint-disable no-undef

REQ-4550: @ts-check  
eslint-env node  
eslint-disable import/no-commonjs

REQ-4551: eslint-env node  
eslint-disable import/no-commonjs

REQ-4552: const rspack = require("@rspack/core");  
const YAML = require("json-to-pretty-yaml");  
const { StatsWriterPlugin } = require("webpack-stats-plugin");  
  
const { WEBPACK\_BUNDLE } =...

REQ-4553: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Account creation",  
 "self": {  
 "vendor": "com.metabase",  
 "name":...

REQ-4554: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Account creation",  
 "self": {  
 "vendor": "com.metabase",  
 "name":...

REQ-4555: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Account creation",  
 "self": {  
 "vendor": "com.metabase",  
 "name":...

REQ-4556: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Action events",  
 "self": {  
 "vendor": "com.metabase",  
 "name":...

REQ-4557: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Schema for tracking clicks on the Browse Data page",  
 "self": {  
 ...

REQ-4558: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Viewing or archiving stale collection items",  
 "self": {  
 "vendor":...

REQ-4559: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Viewing or archiving stale collection items",  
 "self": {  
 "vendor":...

REQ-4560: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "CSV upload events",  
 "self": {  
 "vendor": "com.metabase",  
 "name":...

REQ-4561: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "self": {  
 "vendor": "com.metabase",  
 "name": "csvupload",  
 "format":...

REQ-4562: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "self": {  
 "vendor": "com.metabase",  
 "name": "csvupload",  
 "format":...

REQ-4563: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "self": {  
 "vendor": "com.metabase",  
 "name": "csvupload",  
 "format":...

REQ-4564: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Dashboard events",  
 "self": {  
 "vendor": "com.metabase",  
 "name":...

REQ-4565: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Dashboard events",  
 "self": {  
 "vendor": "com.metabase",  
 "name":...

REQ-4566: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Dashboard events",  
 "self": {  
 "vendor": "com.metabase",  
 "name":...

REQ-4567: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Dashboard events",  
 "self": {  
 "vendor": "com.metabase",  
 "name":...

REQ-4568: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Dashboard events",  
 "self": {  
 "vendor": "com.metabase",  
 "name":...

REQ-4569: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Dashboard events",  
 "self": {  
 "vendor": "com.metabase",  
 "name":...

REQ-4570: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Dashboard events",  
 "self": {  
 "vendor": "com.metabase",  
 "name":...

REQ-4571: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Dashboard events",  
 "self": {  
 "vendor": "com.metabase",  
 "name":...

REQ-4572: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Database connection",  
 "self": {  
 "vendor": "com.metabase",  
 "name":...

REQ-4573: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Database connection",  
 "self": {  
 "vendor": "com.metabase",  
 "name":...

REQ-4574: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Downloads results from questions and dashcards",  
 "self": {  
 "vendor":...

REQ-4575: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "User interactions with public links, static embedding, and public...

REQ-4576: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "User interactions with public links, static embedding, and public...

REQ-4577: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "User interactions with public links, static embedding, and public...

REQ-4578: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "User interactions with public links, static embedding, and public...

REQ-4579: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Schema for tracking embedding enabled and disabled events",  
 "self": {  
 ...

REQ-4580: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Schema for tracking embedding enabled and disabled events",  
 "self": {  
 ...

REQ-4581: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Schema for tracking embedding enabled and disabled events",  
 "self": {  
 ...

REQ-4582: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "User interactions with the embedding-homepage",  
 "self": {  
 "vendor":...

REQ-4583: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Metabase instance",  
 "self": {  
 "vendor": "com.metabase",  
 "name":...

REQ-4584: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Metabase instance",  
 "self": {  
 "vendor": "com.metabase",  
 "name":...

REQ-4585: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Metabase instance",  
 "self": {  
 "vendor": "com.metabase",  
 "name":...

REQ-4586: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Metabase instance",  
 "self": {  
 "vendor": "com.metabase",  
 "name":...

REQ-4587: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Metabase instance",  
 "self": {  
 "vendor": "com.metabase",  
 "name":...

REQ-4588: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Schema for daily stats ping, tracking instance metrics and settings",  
 ...

REQ-4589: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Schema for daily stats ping, tracking instance metrics and settings",  
 ...

REQ-4590: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "New user invite",  
 "self": {  
 "vendor": "com.metabase",  
 "name":...

REQ-4591: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "New user invite",  
 "self": {  
 "vendor": "com.metabase",  
 "name":...

REQ-4592: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Metabot events",  
 "self": {  
 "vendor": "com.metabase",  
 "name":...

REQ-4593: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Metabot events",  
 "self": {  
 "vendor": "com.metabase",  
 "name":...

REQ-4594: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Metabot events",  
 "self": {  
 "vendor": "com.metabase",  
 "name":...

REQ-4595: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Model events",  
 "self": {  
 "vendor": "com.metabase",  
 "name":...

REQ-4596: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "New question creation",  
 "self": {  
 "vendor": "com.metabase",  
 ...

REQ-4597: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "New question creation",  
 "self": {  
 "vendor": "com.metabase",  
 ...

REQ-4598: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "New question creation",  
 "self": {  
 "vendor": "com.metabase",  
 ...

REQ-4599: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Events related to questions",  
 "self": {  
 "vendor": "com.metabase",  
 ...

REQ-4600: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Events related to questions",  
 "self": {  
 "vendor": "com.metabase",  
 ...

REQ-4601: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Events related to questions",  
 "self": {  
 "vendor": "com.metabase",  
 ...

REQ-4602: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Events related to questions",  
 "self": {  
 "vendor": "com.metabase",  
 ...

REQ-4603: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Search events",  
 "self": {  
 "vendor": "com.metabase",  
 "name":...

REQ-4604: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Search events",  
 "self": {  
 "vendor": "com.metabase",  
 "name":...

REQ-4605: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Search events",  
 "self": {  
 "vendor": "com.metabase",  
 "name":...

REQ-4606: {  
 "$schema":"http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description":"Serialization operation",  
 "self":{  
 "vendor":"com.metabase",  
 ...

REQ-4607: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Serialization operation",  
 "self": {  
 "vendor":...

REQ-4608: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Settings events",  
 "self": {  
 "vendor": "com.metabase",  
 "name":...

REQ-4609: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Settings events",  
 "self": {  
 "vendor": "com.metabase",  
 "name":...

REQ-4610: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Settings events",  
 "self": {  
 "vendor": "com.metabase",  
 "name":...

REQ-4611: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Setup flow",  
 "self": {  
 "vendor": "com.metabase",  
 "name":...

REQ-4612: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Setup flow",  
 "self": {  
 "vendor": "com.metabase",  
 "name":...

REQ-4613: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Setup flow",  
 "self": {  
 "vendor": "com.metabase",  
 "name":...

REQ-4614: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Setup flow",  
 "self": {  
 "vendor": "com.metabase",  
 "name":...

REQ-4615: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Generic event to track interactions and actions that happen within...

REQ-4616: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Task history",  
 "self": {  
 "vendor": "com.metabase",  
 "name":...

REQ-4617: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Event timelines",  
 "self": {  
 "vendor": "com.metabase",  
 "name":...

REQ-4618: {  
 "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",  
 "description": "Upsell events",  
 "self": {  
 "vendor": "com.metabase",  
 "name":...

REQ-4619: DO NOT ADD ANY NEW `metabase.x` NAMESPACES HERE. ADD THEM TO THE APPROPRIATE MODULES.  
  
NEW FOLDERS (MODULES) ARE OK!  
  
PLEASE RE-READ...

REQ-4620: DO NOT ADD ANY NEW NAMESPACES WITH API ENDPOINTS HERE. ADD THEM TO THE APPROPRIATE MODULES.  
  
PLEASE RE-READ...

REQ-4621: `channel` module

REQ-4622: DO NOT ADD ANY COMMAND NAMESPACES HERE. ADD THEM TO THE APPROPRIATE MODULES.  
  
PLEASE RE-READ https://www.notion.so/metabase/Guide-to-Backend-Module-Organization-19169354c9018046ab46e4234aace905...

REQ-4623: ---  
title: "Metabase config file template"  
---  
  
# Metabase config file template  
  
You can generate this doc page by changing into the top-level Metabase directory and running:  
  
```  
clojure -M:doc:ee...

REQ-4624: ```

REQ-4625: ---  
title: Environment variables  
redirect\_from:  
 - /docs/latest/operations-guide/environment-variables  
---  
  
# Environment variables  
  
\_This documentation was generated from source by...

REQ-4626: # Other environment variables

REQ-4627: DO NOT ADD ANY NEW EVENT HANDLER NAMESPACES HERE. ADD THEM TO THE APPROPRIATE MODULES.  
  
PLEASE RE-READ...

REQ-4628: DO NOT ADD ANY NEW MODEL NAMESPACES HERE. ADD THEM TO THE APPROPRIATE MODULES.  
  
PLEASE RE-READ https://www.notion.so/metabase/Guide-to-Backend-Module-Organization-19169354c9018046ab46e4234aace905...

REQ-4629: Introduction to Notification

REQ-4630: `metabase.server.\*`

REQ-4631: `setup` Module

REQ-4632: `sync` Module

REQ-4633: DO NOT ADD ANY NEW TASK NAMESPACES HERE. ADD THEM TO THE APPROPRIATE MODULES.  
  
PLEASE RE-READ https://www.notion.so/metabase/Guide-to-Backend-Module-Organization-19169354c9018046ab46e4234aace905...

REQ-4634: Quartz Scheduler in Metabase

REQ-4635: Driver Testing Certificates

REQ-4636: airgap\_eyJhbGciOiJSU0EtT0FFUCIsImVuYyI6IkExMjhDQkMtSFMyNTYifQ.rHUDSPzcF-Kth1pIXbnlLlJiEgwx2rABSyj9PU98ubYGa7vUFwxfR9f4-UhF6EYrjvbtOx0QG9rFOSlfDmgG8jhCpIyuXpfliXdeeFpYt8Pludf-bmB85gU6hQEpYaTq\_hihrPwCjp...

REQ-4637: ssh test keys

REQ-4638: -----BEGIN OPENSSH PRIVATE...

REQ-4639: -----BEGIN RSA PRIVATE...

REQ-4640: -----BEGIN OPENSSH PRIVATE...

REQ-4641: test certificates

REQ-4642: Mongo server with SSL support  
# Running the server

REQ-4643: Oracle certificates