Bundle fixtures

A collection of “smoke”-test that verify that the package layout is correct.

`createFixture` is used to create new or update existing fixtures. The created file might need some manual adjustment since not every edge case is covered.

Run a fixture

1. Navigate into the fixture you want to test (where the `package.json` is located)
2. Use the node version you want to use (e.g. `nvm use 14.0.0`)
3. Prepare the package.json
   - to test a Pull Request
     1. checkout branch
     2. `yarn`
     3. `yarn lerna run build --scope "@mui/*"`
     4. `cd` to fixture
     5. `yarn install`
     6. `node ../../scripts/useBuildFromSource.js .`
   - to test a published npm dist tag (e.g. `latest` or `next`) on npm
     1. `cd` to fixture
     2. adjust the dependencies in the package.json accordingly
     3. `yarn install`
4. `yarn start` should exit with 0

In CI

You have to run our CircleCI pipeline with the `workflow` parameter set to `bundling`.

With the following API request we’re triggering a run of the bundling workflow in PR #24289:

```
curl --request POST \
  --url https://circleci.com/api/v2/project/gh/mui/material-ui/pipeline \
  --header 'content-type: application/json' \
  --header 'Circle-Token: $CIRCLE_TOKEN' \
  --data-raw '{"branch":"pull/24289/head","parameters":{"workflow":"bundling"}}'
```

`$CIRCLE_TOKEN` must be set as an environment variable created from https://app.circleci.com/settings/user/tokens.

Add a new fixture

1. Create a folder in `test/fixtures/bundling`
2. Add the necessary dependencies

3. Re-use the entries for `dependencies` and `resolutions` for `@mui/*` packages from the other fixtures
4. Create a template
5. Write a factory that fills the template in `test/bundling/scripts/createFixture`
6. Add an entry into the `bundling` CircleCI pipeline (`.circleci/config.yml`)