QUASAR FOR ENTERPRISE SOFTWARE

TypeScript, Jest and Cypress
integration without the hassle

Dreamonkey

WHO I AM

Paolo Caleffi

- Class 1992, Italy
- Computer engineer

- Dreamonkey founder
- Full-stack developer
- Quasar core team member, TypeScript support

Dreamonkey

DREAMONKEY IN SHORT

Dreamonkey Srl
develops web apps
and digital integrations
for companies.

We put effort into creating
a more evolved world,
where technology supports
companies, designing
**performant, reliable, simple
and enjoyable web apps,**
which make managers'
and emplyees' lives better.

Dreamonkey

OUR USE CASE

- Dreamonkey works for enterprises

- Enterprises-level software benefit from the usage of:

  – **TypeScript** (compile-time type checking);

  – **Jest** (unit tests);

  – **Cypress** (e2e tests).

Drea**m**onkey

TYPESCRYPT

TS TypeScript

WHY?

- Static type checking

- Static types

- ... but *flexible*

- ... and help avoiding typos

Dreamonkey

TS TypeScript

WHAT'S NEW?

End of 2019: – UI typings generated from JSON API.

Mid 2020:
– Added typings related to **@quasar/app** (boot, configure, etc wrapper functions);
– **Compatible TypeScript version** provided by @quasar/app;
– **Starter kit integration with Composition, Class or Options API:** choose your preferred!
– ... and much more!

Next steps:
– Progressive migration of UI codebase to TS;
– Documentation extracted from TSDocs.

Dreamonkey
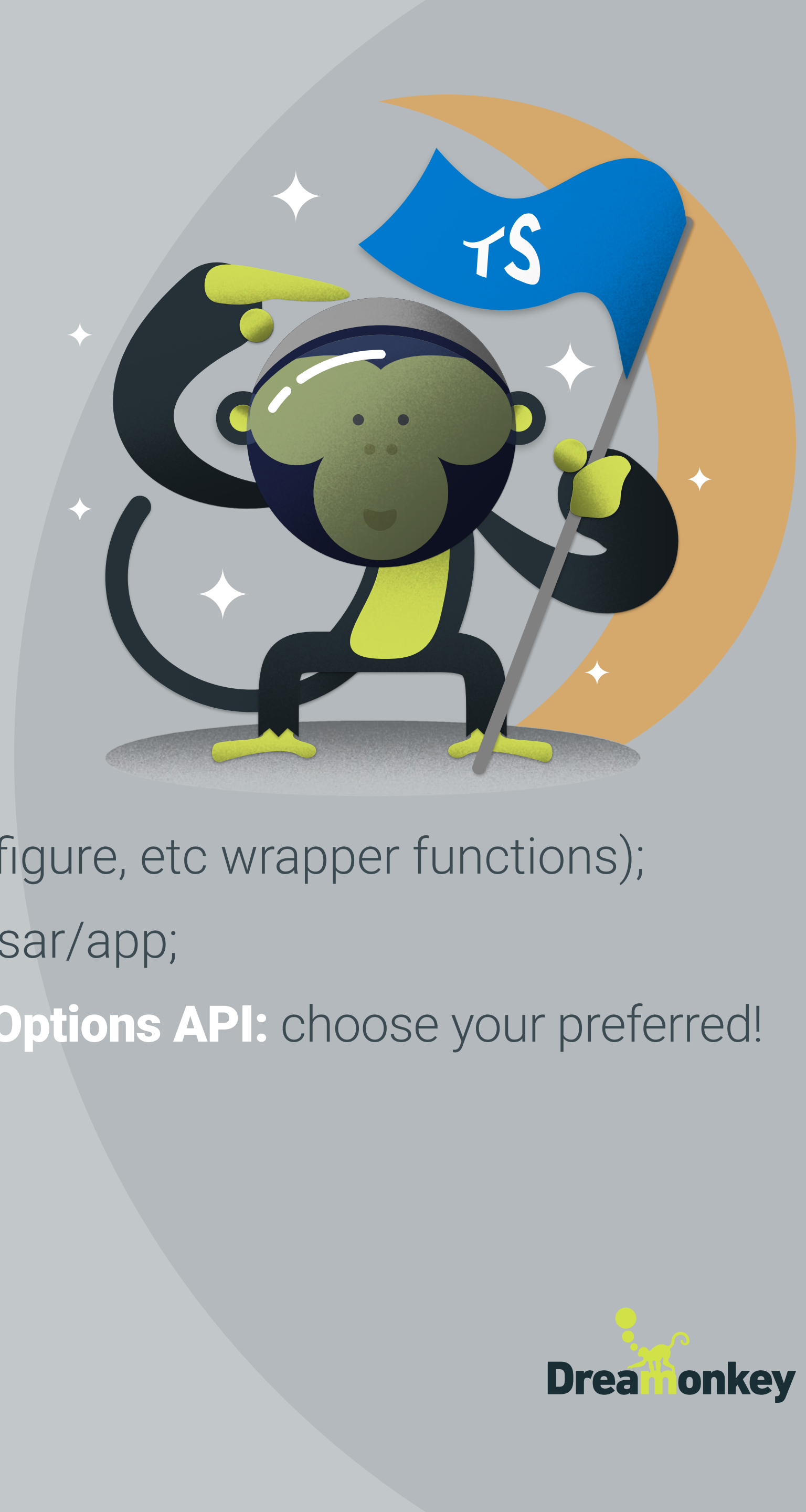
TS TypeScript

HOW TO?

■ New projects:

`quasar create my-ts-project`
and chose `TypeScript support` option

■ Existing projects
(automatic):

ENSURE YOU CAN ROLLBACK
checkout the docs [link]
`quasar ext add @quasar/typescript`
manual adjustments (expect some bumps)

Dreamonkey

TS TypeScript

▌ Existing projects (manual/ enhance current project):

‒ `supportTS: true` into `quasar.config.js`;
‒ create `tsconfig.json` as indicated into docs
https://quasar.dev/quasar-cli/supporting-ts.

▌ Existing projects (manual/start anew):

‒ Create new project;
‒ Manually copy over needed files;
‒ Pray your preferred deity.

▌ TypeScript official documentation:

https://www.staging-typescript.org/docs/home

Dreamonkey

JEST

Jest

WHY?

■ Unit tests

■ Improve stability at component level

■ Spot edge cases in advance

■ Improve overall code quality

Dreamonkey

Jest

WHAT'S NEW?

- TypeScript support **(beta)**

- Revamped and fully typed "mountQuasar" helper **(soon to come)**

Dreamonkey

Jest

HOW TO?

```
quasar ext add @quasar/testing-unit-jest@beta
```

Jest official documentation

https://jestjs.io/docs/en/getting-started

Dreamonkey

cy Cypress

WHY?

- e2e tests

- Guarantee stability of critical features at system level

- Classical example: perform login, navigate all pages, perform logout

Dreamonkey

cy **Cypress**

HOW TO?

```
Quasar ext add @quasar/testing-e2e-cypress@beta
```

Cypress official documentation

https://docs.cypress.io/guides/overview/why-cypress.html#In-a-nutshell

Dreamonkey

TODAY WE LEARNED...

- TypeScript

- Jest

- Cypress

- We manage the integration / you learn and use the techs you like

DO YOU <3 QUASAR AS WE DO?

Start a monthly donation

With $$ (pretty-please your boss for this)

With your time (aka "help us fixing bugs and creating repros")

Sponsor a new feature's development

Dreamonkey

Evolution in motion

Dreamonkey Srl
P. IVA/CF 02722510357
via Mazzacurati 3B
42019 Scandiano [RE]
Italy

www.dreamonkey.com
info@dreamonkey.com
+39 348 663 8611

SO LONG AND THANKS FOR ALL THE FISH

Q & A