

[< VOLTAR](#)

Gerenciamento de estados com MobX

Nessa aula vamos aprender um recurso muito usado no mercado de aplicações profissionais para gerenciamento de estados da aplicação. O famoso, MobX, que nasceu para o Javascript, mas chegou recentemente ao Flutter e impressionou a todos os desenvolvedores com a facilidade de uso e desempenho.

NESTE TÓPICO

- › MobX
- › Desenvolvimento da aplicação MobX
- › Referências



MobX



O MobX é uma biblioteca de gestão de estados que nasceu originalmente para o desenvolvimento híbrido com Javascript, especialmente para uso com React Native, mas tornou-se tão importante para o desenvolvimento de aplicativos profissionais que os desenvolvedores criaram recentemente uma versão para o Flutter.

Qualquer coisa que possa ser derivada do estado do aplicativo deve ser. Automaticamente.

(<https://mobx.js.org/README.html>)

Para entendermos um pouquinho melhor de como o MobX funciona, vamos assistir a breve apresentação abaixo.

MobX com Flutter



CONHEÇA MAIS O MOB X

Site original do MobX para JS: <https://mobx.js.org/README.html>
(<https://mobx.js.org/README.html>)

Site oficial do MobX para Flutter: <https://pub.dev/packages/mobx>
(<https://pub.dev/packages/mobx>)



Desenvolvimento da aplicação MobX

Vamos desenvolver o projeto de um pequeno jogo de dados utilizando o MobX e exemplos fornecidos pela documentação oficial do Flutter, portanto, vamos implementar um código Open Source (de código aberto) baseado no site oficial.

Para isso, algumas imagens serão necessárias como recursos deste projeto e você poderá realizar o download delas no link abaixo. Essas imagens são as faces dos dados que serão utilizadas na visualização dos resultados.

MATERIAL COMPLEMENTAR

(<https://img.uninove.br/static/0/0/0/0/0/9/2/7/1/1/9271174/62498-Imagens-Dados.7z>)

Com as imagens em mãos, vamos começar o desenvolvimento de nosso jogo. Para esse projeto usamos várias dependências externas em nosso *pubspec.yaml*, conforme:

1. provider: ^4.3.3
2. mobx: ^1.2.1+4
3. flutter_mobx: ^1.1.0+2

E utilizamos também algumas dependências de desenvolvimento, conforme:

1. dev_dependencies: //não é preciso incluir esta linha
2. mobx_codegen: ^1.1.2
3. build_runner: ^1.11.0

Recomenda-se fortemente que você utilize as mesmas versões para não ter nenhum problema de compatibilidade com o MobX, pois essa biblioteca tem sofrido muitas atualizações constantes.

O vídeo inicial do desenvolvimento da aplicação explica melhor onde colocar cada uma dessas dependências, conforme abaixo.

Codegen do MobX



Durante o desenvolvimento desta aula, usamos um comando de terminal que você pode ver abaixo, caso necessário para consulta. Lembre-se que este comando deve ser executado com seu terminal (prompt de comando) dentro da pasta do projeto em questão.

1. flutter pub run build_runner build

Agora que a base da aplicação já está pronta e o MobX já está integrado ao nosso objeto observado, podemos começar a criar a interface de usuário. Sim, dessa vez fizemos toda a lógica da aplicação e a interface por último.

O vídeo abaixo mostra, portanto, a criação da interface do usuário com integração a este projeto e mostra a realização de alguns testes.

Finalização do Jogo



E, caso você precise do projeto completo para consulta, basta fazer o download no link abaixo. É importante mencionar que o projeto abaixo já possui o código gerado e as imagens de recursos.

MATERIAL COMPLEMENTAR

(<https://img.uninove.br/static/0/0/0/0/0/0/9/3/1/5/8/9315821/62501.7z>)



Prontinho, chegamos ao final de mais uma importante aula com importante conceito aprendido, o MobX. Esperamos de coração que as aulas de Flutter tenham sido muito proveitosas e que lhe ajudem a trilhar um caminho, indubitavelmente, de sucesso.

Quiz

Exercício Final

Gerenciamento de estados com MobX

INICIAR ➤

Referências

DART. **Dart documentation**. *Site*. Disponível em: <https://dart.dev/>. Acesso em: 08 dez. 2020.

MATERIAL DESIGN. **Material Design documentation**. *Site*. Disponível em: <https://material.io/>. Acesso em: 08 dez. 2020.

FLUTTER. **Flutter docs**. *Site*. Disponível em: <https://flutter.dev>. Acesso em: 09 dez. 2020.

WINDMILL, Eric. **Flutter in action**. Nova Iorque: Manning publications, 2020. *E-book*. Disponível em: <https://learning.oreilly.com/library/view/flutter-in-action/9781617296147/>. Acesso em: 08 dez. 2020.

SINHA, Sanjib. **Quick start guide to Dart programming**: create high performance applications for the web and mobile. Lompoc, CA, EUA: Apress, 2019. *E-book*. Disponível em: <https://learning.oreilly.com/library/view/quick-start-guide/9781484255629/>. Acesso em: 09 dez. 2020.

ALESSANDRIA, Simone. **Flutter projects**: a practical, project-based guide to building real-worlds cross-platform mobile applications and games. Birmingham, Reino Unido: Packt Publishing, 2020. *E-book*. Disponível em: <https://learning.oreilly.com/library/view/flutter-projects/9781838647773/>. Acesso em: 09 dez. 2020.

ZACCAGNINO, Carmine. **Programming Flutter**. [s.l.]: The Pragmatic Bookshelf, 2020. *E-book*. Disponível em: <https://learning.oreilly.com/library/view/programming-flutter/9781680507621/>. Acesso em: 09 dez. 2020.



Avalie este tópico



ANTERIOR

Integração com código nativo



Índice

Biblioteca
(<https://www.uninove.br/conheca-a-uninove/biblioteca/sobre-a-biblioteca/apresentacao/>)
Portal Uninove
(<http://www.uninove.br>)
Mapa do Site

® Todos os direitos reservados

Ajuda?
(<https://ava.uninove.br/ava/uninove/idCurso=>)

