Match Made in Heaven NgRx Signal Store

ANGULAR
ARCHITECTS

Rainer Hahnekamp - 10. February 2024 - Ng-India

About Me...

Experts
Angular

- Rainer Hahnekamp

  ANGULARarchitects.io

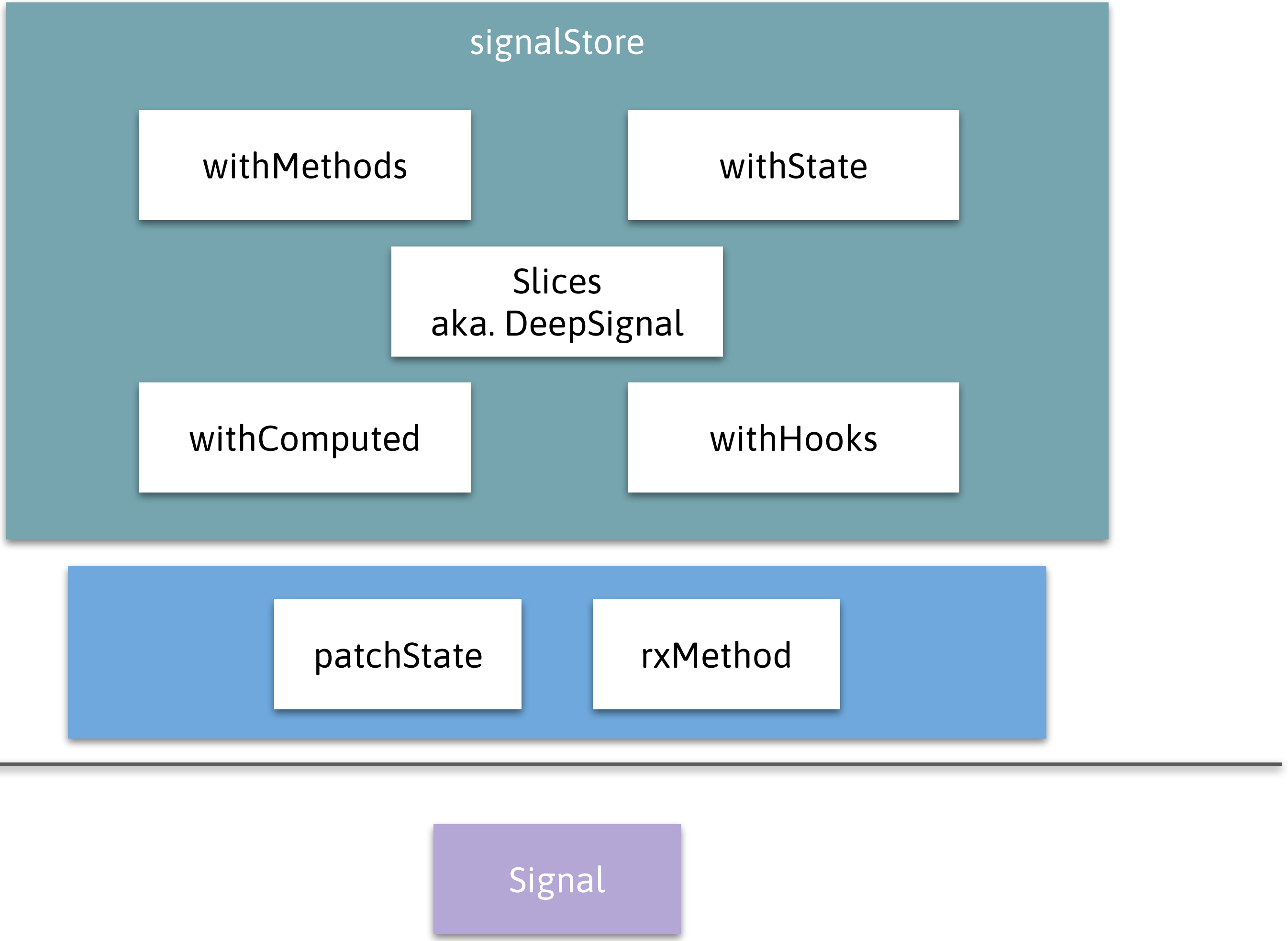
- Developer / Trainer / Speaker

https://www.youtube.com/
@RainerHahnekamp

https://www.ng-news.com

https://github.com/softarc-consulting/sheriff

@RainerHahnekamp


signalStore
withMethods
withState
Slices
aka. DeepSignal
withComputed
withHooks
patchState
rxMethod
Signal

`withState({someState})`

- Contains the State internally as a Signal
- Exposes Signal Slices
- Allows easy updates via patchState

signalStore & withState(*{someState}*)

```
export const QuizStore = signalStore(
  withState({
    title: '',
    questions: [] as Question[],
    timeInSeconds: 60
  })
)
```

`withMethods(() => {methods})`

- Adds public methods to the Store
- Binds Data (State) and Logic (Methods) together
- Provides Access to DI

withMethods(() => {*methods*})

```
withMethods((store) => {
  const quizService = inject(QuizService);

  return {
    answer(questionId: number, choiceId: number) {
      const question = store
        .questions()
        .find((question) => question.id === questionId);
      assertDefined(question);

      patchState(store, (quiz) => ({
        // ...
      }));
   }
})
```

`withComputed(() => {`*`computeds`*`})`

- Adds derived Signals
- Collection of multiple `computed()` in one place

withComputed(() => {*computeds*})

```
withComputed((state) => {
  return {
    status: computed(() => {
      const status: Record<AnswerStatus, number> = {
        unanswered: 0,
        correct: 0,
        incorrect: 0,
      };

      for (const question of state.questions()) {
       status[question.status]++;
      }

      return status;
    }),
  };
})
```

rxMethod: Integrating RxJs

```
withMethods((store) => {
  const quizService = inject(QuizService);
  return {
    setId: rxMethod<number>(
      pipe(
        switchMap((id) => quizService.findById(id)),
        tap((quiz) => patchState(store, quiz)),
      ),
    ),
}))
```

Summary

- **Embraces** (builds upon) Angular's Signal
  - Reactivity
  - Computed
  - Immutability
- **Extends** the Signal
  - patchState
  - Slices
- Adds support for **asynchronous + Signals**
- Optionally integrates **RxJs**
- Brings Logic and Data in a **structured** way together
- Highly **Extensible**
- Provides **Local** and **Global State**

Try it out!

RainerHahnekamp

Introducing NgRx Toolkit

@angular-architects/ngrx-toolkit

Redux

Entities Extensions with API
@ngrx/data

DevTools

Encapsulation

Further Reading and Watching

https://youtu.be/V-D2sk_azcs

https://youtu.be/TLq0OcSshYI

Further Reading and Watching

- https://ngrx.io/guide/signals
- https://medium.com/ngconf/ngrx-signal-store-the-missing-piece-to-signals-ac125d804026
- https://www.angulararchitects.io/en/blog/the-new-ngrx-signal-store-for-angular-2-1-flavors
- https://www.youtube.com/watch?v=yaOLbKwVRtc

धन्यवाद्

dhanyavaad

𝕏 RainerHahnekamp