

KeepAlive

<KeepAlive> 는 여러 컴포넌트 간에 동적으로 전환될 때, 컴포넌트 인스턴스를 조건부로 캐시할 수 있는 빌트인 컴포넌트입니다.

기본 사용법

컴포넌트 기초의 동적 컴포넌트 장에서 특별한 <component> 엘리먼트를 사용하는 문법을 소개했습니다:

```
<component :is="activeComponent" />
```

template

기본적으로 활성 컴포넌트 인스턴스는 전환할 때 마운트 해제됩니다. 이렇게 하면 해당 컴포넌트가 보유한 모든 변경된 상태가 손실됩니다. 이 컴포넌트가 다시 표시되면 초기 상태로만 새 인스턴스가 생성됩니다.

아래 예시에는 두 개의 상태 저장 컴포넌트가 있습니다. A에는 카운터가 있고, B에는 v-model 을 통해 입력과 동기화된 메시지가 있습니다. 이 중 하나의 상태를 업데이트하고 다른 곳으로 전환했다가 다시 전환해 보세요:

☒ A ☐ B

Current component: A

Count: 0 +

다시 전환하면 이전에 변경한 상태가 초기화되었음을 알 수 있습니다.

스위치에서 새 컴포넌트 인스턴스를 만드는 것은 일반적으로 유용한 동작이지만 이 경우에는 두 컴포넌트 인스턴스가 비활성 상태인 경우에도 상태가 보존되기를 원합니다. 이 문제를 해결하기 위해 동적 컴포넌트를 빌트인 컴포넌트인 <KeepAlive> 로 래핑할 수 있습니다:

```
<!-- 비활성 컴포넌트가 캐시됩니다! -->
<KeepAlive>
  <component :is="activeComponent" />
</KeepAlive>
```

template

이제 상태는 컴포넌트 전환 간에 유지됩니다:

☒ A ☐ B

Current component: A

Count: 0 +

온라인 연습장으로 실행하기

TIP

in-DOM 템플릿에서 사용할 때 <keep-alive> 로 참조해야 합니다.

Include / Exclude

기본적으로 <KeepAlive> 는 내부의 모든 컴포넌트 인스턴스를 캐시합니다. include 및 exclude props를 통해 이 동작을 사용자 정의할 수 있습니다. 두 props 모두 선택으로 구분된 문자열, RegExp (정규식) 또는 이 두 유형 중 하나를 포함하는 배열이 될 수 있습니다:

```

<!-- 쉼표로 구분되는 문자열 -->
<KeepAlive include="a,b">
  <component :is="view" />
</KeepAlive>

<!-- 정규식 (`v-bind`를 사용해야 함) -->
<KeepAlive :include="/a|b/">
  <component :is="view" />
</KeepAlive>

<!-- 배열 (`v-bind`를 사용해야 함) -->
<KeepAlive :include="['a', 'b']">
  <component :is="view" />
</KeepAlive>

```

컴포넌트의 **name** 옵션과 일치하는지 확인하므로, `KeepAlive` 에 의해 조건부로 캐시되어야 하는 컴포넌트는 명시적으로 **name** 옵션을 선언해야 합니다.

TIP

버전 3.2.34부터 `<script setup>` 을 사용하는 싱글 파일 컴포넌트는 파일명을 기반으로 **name** 옵션을 자동으로 추론하므로 이름을 수동으로 선언할 필요가 없습니다.

최대 캐시 인스턴스

`max props`를 통해 캐시할 수 있는 컴포넌트 인스턴스의 최대 수를 제한할 수 있습니다. `max` 가 지정되면 `<KeepAlive>` 는 **LRU** 캐시처럼 작동합니다. 캐시된 인스턴스의 수가 지정된 최대 수를 초과하려고 하면, 가장 최근에 접근해서 캐시된 인스턴스가 파괴되어 새 인스턴스를 위한 공간을 확보합니다.

```

<KeepAlive :max="10">
  <component :is="activeComponent" />
</KeepAlive>

```

캐시된 인스턴스의 생명주기

컴포넌트 인스턴스가 DOM에서 제거되었지만 `<KeepAlive>` 에 의해 캐시된 컴포넌트 트리의 일부인 경우, 마운트 해제되는 대신 **비활성화**된 상태가 됩니다. 컴포넌트 인스턴스가 캐시된 트리의 일부로 DOM에 삽입되면 **활성화**됩니다.

`kept-alive` 컴포넌트는 `onActivated()` 및 `onDeactivated()` 를 사용하여 이 두 가지 상태에 대한 생명 주기 훅을 등록할 수 있습니다:

```

<script setup>
import { onActivated, onDeactivated } from 'vue'

onActivated(() => {
  // 초기 마운트 시 또는
  // 캐시상태에서 다시 삽입될 때마다 호출됨.
})

onDeactivated(() => {
  // DOM에서 제거되고 캐시로 전환될 시 또는
  // 마운트 해제될 때마다 호출됨.
})

```

```
})  
</script>
```

참고:

`onActivated` 는 마운트 시에도 호출되고 `onDeactivated` 는 마운트 해제 시에도 호출됩니다.

두 혹은 `<KeepAlive>` 에 의해 캐시된 루트 컴포넌트뿐만 아니라 캐시된 트리 내의 하위 컴포넌트에도 작동합니다.

관련 문서

`<KeepAlive>` API 참고