Vue lifecycle hooks explained

Vue lifecycle hooks: A set of lifecycle hooks are offered by the well-known JavaScript framework Vue to assist you in managing and controlling the behavior of your components. At various times during a component's lifespan, these hooks are functions that are invoked.

Here is a quick breakdown of the various Vue lifecycle hooks:

1. **Before Create**: Before any data or methods are set up, the first hook, before Create, is called whenever a new component instance is created.
2. **Created**: After the component has been formed and all data and methods have been configured, this hook is invoked. The component's template has not yet been assembled at this time.
3. **BeforeMount**: Before the component is mounted to the DOM, this hook is called before Mount. The feature is now prepared to be added to the document because the template has been put together.
4. **mounted**: After the component has been mounted to the DOM, this hook is invoked. Any initialization that needs access to the DOM can be completed here.
5. **Before update**: The component is updated, whether as a result of changes to its data or props, this hook is invoked.
6. **updated**: After the component has been updated, this hook is invoked. The component's DOM has now been updated to reflect the modifications to its data or props.
7. **Before unmounted**: The component is unmounted from the DOM, this hook is triggered.

8. **Unmounted**: After the component has been demounted from the DOM, this hook is called. Any clean-up work that needs to be done can be done here.

In addition, there are a few less popular hooks that can be helpful in certain particular circumstances. These include enabled, deactivated, error Captured, and render Tracked. It's crucial to comprehend the Vue lifecycle hooks if you want to develop applications that are stable and have a clear structure. Using these hooks, you may influence how your components behave at various points in their lifetime, which will help your code operate smoothly and effectively.

To know more: Vue Js Lifecycle Explained: From Initialization To Destruction