# AppRegistry

AppRegistry is the control point for registering, running, prerendering, and unmounting all apps.

App root components should register themselves with `AppRegistry.`registerComponent. Apps can be run by invoking `AppRegistry.runApplication`.

```
import { AppRegistry } from 'react-native';
```

## API

### Static methods

> **getAppKeys**: () => Array<string>
> Returns an array of all registered app keys

> **getApplication**: (key: string, params: AppParams) => ({ element, getStyleElement })
> A web-only method for server-side rendering to HTML and CSS. It returns an object containing the given application's `element` and `getStyleElement` function to get styles once the element is rendered.

> **registerComponent**: (key: string, getComponent: () => React.Element) => void
> Register a component provider under the given `key`.

> **registerConfig**: (configs: Array<AppConfig>) => void
> Register multiple applications.

> **unmountApplicationComponentAtRootTag**: rootTag: HTMLElement
> Called this function with the `rootTag` that was passed into `runApplication` in order to unmount it.

### AppConfig

> **appKey**: string
>
> The `key` under which the component is registered.

> **component**: () => React.Element
>
> A function that returns a React element.

## AppParams

> **callback**: ?() => void
>
> Called when React rendering has finished.

> **hydrate**: ?boolean
>
> If the client should hydrate server-rendered HTML.

> **initialProps**: ?Object
>
> The initial props passed to the root component.

> **mode**: "concurrent" | "legacy"
>
> Default is ‘concurrent’. Setting to ‘legacy’ will make the app will behave as if it’s running React 17.

> **rootTag**: HTMLElement
>
> The native element into which the application is rendered.

Updated July 20, 2023 Edit