Vue.js Fundamentals

GETTING STARTED WITH THE VUE.JS CLI



Jim Cooper
SOFTWARE ENGINEER

@jimthecoop jcoop.io

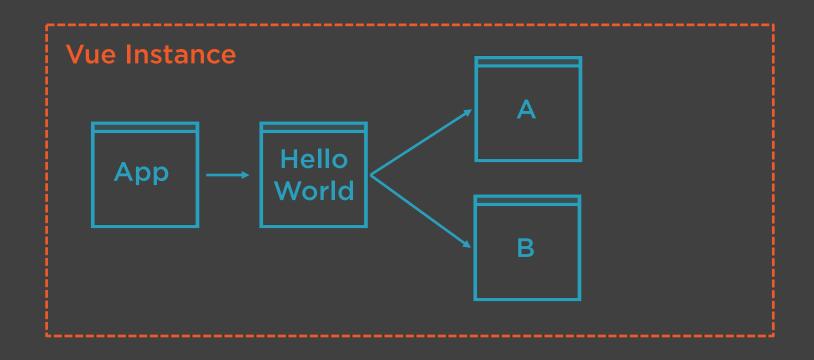


```
import * as Vue from 'vue';
import App from './App.vue';
Vue.createApp(App)
```

Vue.js Application Structure

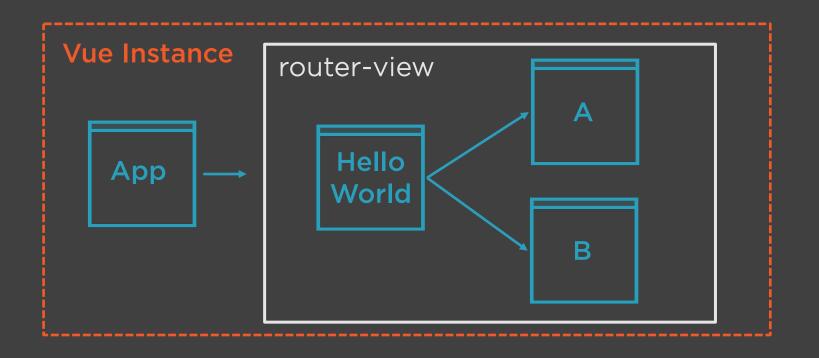
The Vue instance





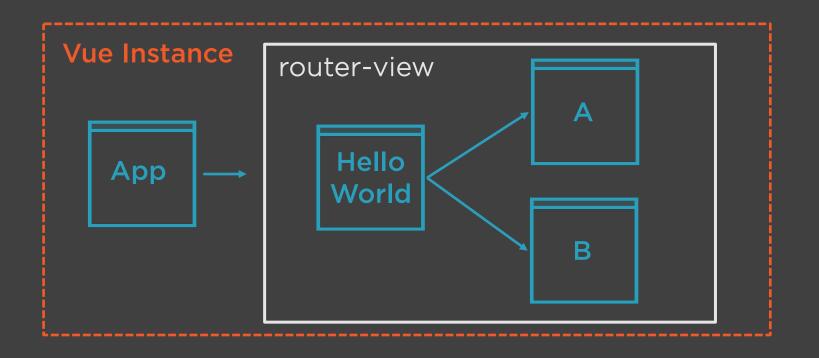
Vue.js Application Structure Component hierarchy





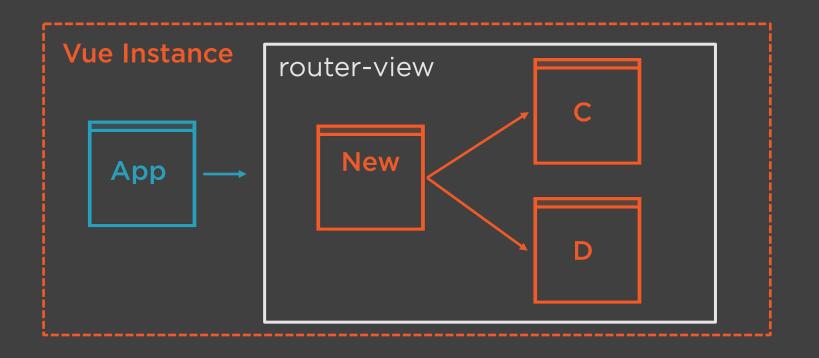
Vue.js Application Structure Routing





Vue.js Application Structure Routing





Vue.js Application Structure Routing



Composition API

Options API

Simple

Easy to see everything at a glance

Larger components become convoluted

Composition API

Allows for abstraction / composition

Splits major concepts into separate files

Easier to understand in complex components



Example: Search Component

```
data() {
                                                             Search
computed: {
                                                              Filter
methods: {
                                                           Pagination
```



Example: Search Component

```
data() {
                                                              Search
computed: {
                                                              Filter
methods: {
                                                            Pagination
```



```
<script>
export default {
 setup() {
    let searchResults = searchInventory();
   const search = () => { ... };
    return {
      resultCount: searchResults.length,
      search,
```



```
<script>
export default {
  props: { searchTerm: { type: String } },
  components: { SearchResultsItem },
  setup(props) {
    let searchResults = searchInventory(props.searchTerm);
    const search = () => { ... };
    return {
      resultCount: searchResults.length,
      search,
</script>
```

Lifecycle Hooks

Options API

beforeCreate created beforeMount mounted beforeUpdate updated beforeUnmount unmounted errorCaptured renderTracked renderTriggered activated deactivated

Composition API

Not Needed *Not Needed* onBeforeMount onMounted onBeforeUpdate onUpdated onBeforeUnmount onUnmounted onErrorCaptured onRenderTracked onRenderTriggered onActivated onDeactivated



```
import { onMounted } from 'vue';
setup(props) {
 let results = searchInventory(props.searchTerm);
  const search = () => { ... };
  const handleMounted = () => { ...};
  onMounted(handleMounted);
  return {
    resultCount: results.length,
    search,
```

```
setup(props) {
 let currentPage = 1;
  const handlePagingMount = () => { ... };
  const handleMount = () => {
    handleSearchMount();
    handlePagingMount();
 onMounted(handleMount);
  return {
   filters,
    currentPage,
    applyFilters,
   nextPage,
    prevPage
```

Search

Filter

Pagination



```
setup(props) {
  let results = searchInventory(props.searchTerm);
  let filters = { };
  let currentPage = 1;
  const search = () => { ... };
  const applyFilters = () => { ... };
  const nextPage = () => { ... };
  const prevPage = () => { ... };
  const handleSearchMount = () => { ... };
  const handleFilterMount = () => { ... };
  const handlePagingMount = () => { ... };
  const handleMount = () => {
    handleSearchMount();
    handleFilterMount();
    handlePagingMount();
  onMounted(handleMount);
  return {
    resultCount: results.length,
    filters,
    currentPage,
    search,
    applyFilters,
    nextPage,
    prevPage
```

```
setup(props) {
 let currentPage = 1;
  const handlePagingMount = () => { ... };
  const handleMount = () => {
    handleSearchMount();
    handlePagingMount();
 onMounted(handleMount);
  return {
   filters,
    currentPage,
    applyFilters,
   nextPage,
    prevPage
```

Search

Filter

Pagination



```
setup(props) {
 let filters = { };
 let currentPage = 1;
                                                        onMounted(() => { ... });
  const handlePagingMount = () => { ... };
                                                        let filters = { };
  const handleMount = () => {
    handleSearchMount();
   handleFilterMount();
   handlePagingMount();
                                                        return { filters, applyFilters };
 onMounted(handleMount);
                                                      export default function usePagination() {
                                                        let currentPage = 1;
  return {
    resultCount: results.length,
                                                        const nextPage = () => { ... };
                                                        const prevPage = () => { ... };
   filters,
   currentPage,
                                                        onMounted(() => { ... });
   applyFilters,
                                                        return { currentPage, nextPage, prevPage };
   nextPage,
   prevPage
```

```
let results = searchInventory(props.searchTerm);
import usePagination from './usePagination';
                                                       onMounted(() => { ... });
setup(props) {
                                                       let filters = { };
                                                       return { filters, applyFilters };
                                                     export default function usePagination() {
                                                       let currentPage = 1;
                                                       const nextPage = () => { ... };
                                                       const prevPage = () => { ... };
                                                       onMounted(() => { ... });
                                                       return { currentPage, nextPage, prevPage };
```



```
import usePagination from './usePagination';
setup(props) {
 return {
   ...useFilters(),
   ...usePagination()
```

```
onMounted(() => { ... });
 let filters = { };
  return { filters, applyFilters };
export default function usePagination() {
  let currentPage = 1;
  const nextPage = () => { ... };
  const prevPage = () => { ... };
  onMounted(() => { ... });
  return { currentPage, nextPage, prevPage };
```



```
import useFilters from './useFilters';
import usePagination from './usePagination';
setup(props) {
 const { resultCount, search } = useSearch();
 const { filters, applyFilters } = useFilters();
 const { currentPage, nextPage, prevPage } =
   usePagination();
```

```
onMounted(() => { ... });
export default function useFilters() {
  let filters = { };
  return { filters, applyFilters };
export default function usePagination() {
  let currentPage = 1;
  const nextPage = () => { ... };
  const prevPage = () => { ... };
  onMounted(() => { ... });
  return { currentPage, nextPage, prevPage };
```



```
import usePagination from './usePagination';
setup(props) {
 const { filters, applyFilters } = useFilters();
 const { currentPage, nextPage, prevPage } =
   usePagination();
 return {
   resultCount,
   filters,
   currentPage,
   nextPage,
   prevPage
```

```
onMounted(() => { ... });
export default function useFilters() {
  let filters = { };
  return { filters, applyFilters };
export default function usePagination() {
  let currentPage = 1;
  const nextPage = () => { ... };
  const prevPage = () => { ... };
  onMounted(() => { ... });
  return { currentPage, nextPage, prevPage };
```

