

Dispositivos Portáteis

Genilson Medeiros Martins

MDB em Gerenciamento de Projetos - PUC Minas

Mestre em Tecnologias Estratégicas em Saúde - UEPB/NUTES

Engenheiro de Software / Líder Técnico - VIRTUS/UFCG

Graduado em Sistemas de Informação - UNIFACISA



 ionicframework

ionic.

Storage

<https://github.com/ionic-team/ionic-storage?tab=readme-ov-file#api>



Storage



Um módulo de armazenamento de chave-valor simples para aplicativos Ionic. Este utilitário usa o melhor mecanismo de armazenamento disponível na plataforma sem precisar interagir diretamente com ele



Pronto para uso, o Ionic Storage usará IndexedDB e localStorage quando disponível. Para usar o SQLite para armazenamento nativo, veja as instruções de instalação do SQLite.



Criando o app

ionic start

npm install @ionic/storage

Criando o app

ToDo

Cadastro

ADICIONAR

Atv3

Atv1

Atv2

<https://github.com/ionic-team/ionic-storage?tab=readme-ov-file#api>

Criando o app

```
<ion-card>
  <ion-card-header>
    <ion-card-title>Cadastro</ion-card-title>
  </ion-card-header>
  <ion-card-content>
    <ion-input v-model="input" @keydown.enter="confirm" label="Atividade" label-placement="floating" fill="outline"
      placeholder="Atividade"></ion-input>
    <ion-button @click="confirm" expand="block">Adicionar</ion-button>
  </ion-card-content>
</ion-card>

<ion-card>
  <ion-list>
    <ion-item @click="goToDetail(item.id)" v-for="(item, index) in tasks" :key="index">
      <ion-label>{{ item.activity }}</ion-label>
    </ion-item>
  </ion-list>
</ion-card>

<ion-note color="danger">
  <p class="note"><a href="https://github.com/ionic-team/ionic-storage?tab=readme-ov-file#api">https://github.com/ionic-team/ionic-storage?tab=readme-ov-file#api</a></p>
</ion-note>
```

Criando o app

```
import {
  IonContent,
  IonHeader,
  IonPage,
  IonTitle,
  IonToolbar,
  IonInput,
  IonCard,
  IonCardHeader,
  IonCardTitle,
  IonCardContent,
  IonButton,
  IonNote,
  IonItem, IonLabel, IonList
} from '@ionic/vue';
import { reactive, ref, onMounted } from 'vue';
import { Task } from '@/model/task';
import { Messages } from '@/model/messages';
import { Storage } from '@ionic/storage';
import { useRouter } from 'vue-router';
```

```
const router = useRouter()
const input = ref<string>('')
const tasks = reactive<Task[]>([])
const store = new Storage();
const db_table = 'tasks';
```

Criando o app - Interface

```
export interface Task {  
  id: number;  
  activity: string;  
}
```

```
export class Messages {  
  public static readonly CONFIRM: string = 'Confirmar adição da atividade?';  
  public static readonly VALIDATION: string = 'Campo vazio. Digite o nome de uma atividade!';  
}
```


Criando o app

```
onMounted(async () => {  
  await store.create()  
  await updateList()  
})
```

```
async function updateList() {  
  const db = await store.get(db_table)  
  if(db) {  
    const tableValues = JSON.parse(db)  
    tasks.splice(0, tasks.length)  
    tableValues.forEach((element: Task) => tasks.push(element));  
    tasks.reverse()  
  }  
}
```

Criando o app

```
<ion-card>
  <ion-card-header>
    <ion-card-title>Cadastro</ion-card-title>
  </ion-card-header>
  <ion-card-content>
    <ion-input v-model="input" @keydown.enter="confirm" label="Atividade" label-placement="floating" fill="outline"
      placeholder="Atividade"></ion-input>
    <ion-button @click="confirm" expand="block">Adicionar</ion-button>
  </ion-card-content>
</ion-card>
```

```
async function confirm() {
  if (input.value == '') {
    alert(Messages.VALIDATION)
    input.value = ''
    return
  }

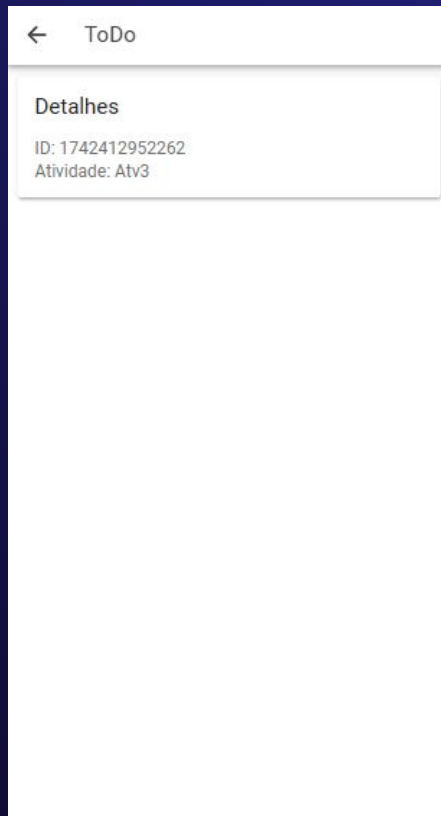
  await add({ id: new Date().getTime(), activity: input.value})
  await updateList()

  input.value = ''
}
```

Criando o app

```
async function add(task: Task) {  
  tasks.push(task)  
  await store.set(db_table, JSON.stringify(tasks))  
}
```

Criando o app - Details



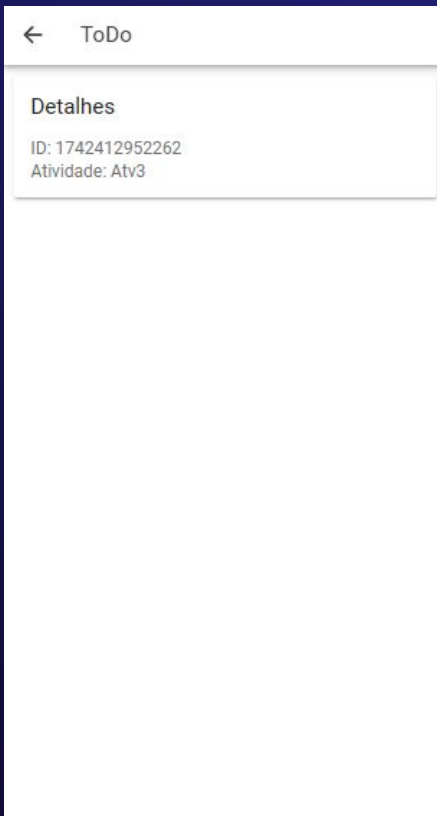
```
<ion-card>
  <ion-list>
    <ion-item @click="goToDetail(item.id)" v-for="(item, index) in tasks" :key="index">
      <ion-label>{{ item.activity }}</ion-label>
    </ion-item>
  </ion-list>
</ion-card>
```

```
import { useRouter } from 'vue-router';

const router = useRouter()
```

```
function goToDetail(index: number) {
  router.push(`/detail/${index}`)
}
```

Criando o app - Details



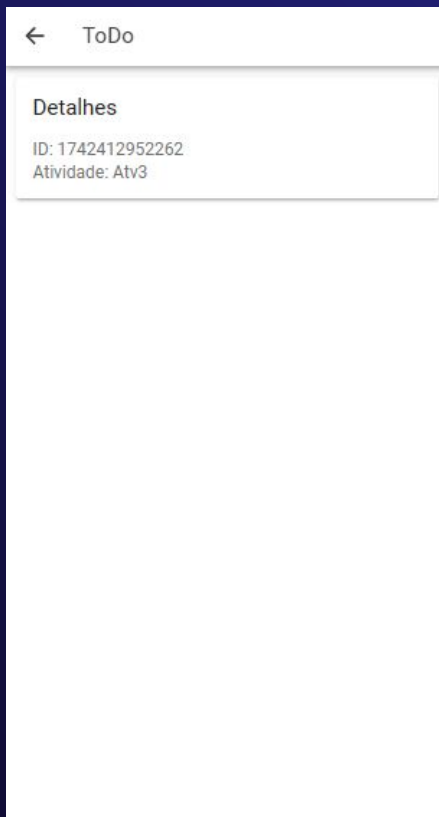
```
import { createRouter, createWebHistory } from '@ionic/vue-router';
import { RouteRecordRaw } from 'vue-router';
import HomePage from '../views/HomePage.vue'
import DetailPage from '@views/DetailPage.vue'

const routes: Array<RouteRecordRaw> = [
  {
    path: '/',
    redirect: '/home'
  },
  {
    path: '/home',
    name: 'Home',
    component: HomePage
  },
  {
    path: '/detail/:id',
    name: 'Detail',
    component: DetailPage
  }
]

const router = createRouter({
  history: createWebHistory(import.meta.env.BASE_URL),
  routes
})

export default router
```

Criando o app - Details



```
<ion-card>
  <ion-card-header>
    <ion-card-title>Detalhes</ion-card-title>
  </ion-card-header>
  <ion-card-content>
    <h2>ID: {{ task?.id }}</h2>
    <h2>Atividade: {{ task?.activity }}</h2>
  </ion-card-content>
</ion-card>
```

```
import { onMounted, ref } from 'vue';
import { useRoute } from 'vue-router' ←

import { Storage } from '@ionic/storage';
import { Task } from '@model/task'; ←

const route = useRoute(); ←
const task = ref<Task>({} as Task)
```

Criando o app - Details

```
onMounted(async () => {  
  const id: number = Number(route.params.id)  
  
  const store = new Storage();  
  await store.create()  
  const db = await store.get('tasks')  
  
  if(db) {  
    const tableValues = JSON.parse(db) as Task[]  
    task.value = tableValues.find(item => item.id == id) as Task  
  }  
})
```


Storage

Application

Manifest

Service workers

Storage

Storage

Local storage

Session storage

Extension storage

IndexedDB

Cookies

Private state tokens

Interest groups

Shared storage

Cache storage

Storage buckets

Storage

http://localhost:8100/

Usage

4.0 kB used out of 599,643 MB storage quota

[Learn more](#)

4.0 kB

4.0 kB

IndexedDB

4.0 kB

Total

☐ Simulate custom storage quota

Storage



Application

- Manifest
- Service workers
- Storage

Storage

- Local storage
- Session storage
- Extension storage
- IndexedDB
- _ionicstorage**
- _ionickv
- local-forage-detect-blob-supp...

Filter by key (show keys greater or equal to)

#	Key	Value
0	"tasks"	▶ "[{"id\":"1742412155470","\activity\":"Atv2\"}, {"id\":"1742412150838","\activity\":"Atv1\"}, {"id\":"1742412952262","\activity\":"Atv3\"}]"

