



## Grupo do Whatsapp para tirar dúvidas:

<https://chat.whatsapp.com/FcojWleXXpDLwZYz7zLVK1>

## Introdução ao Ionic

### O que é um framework?

É um conjunto de bibliotecas que são usadas para criar uma base onde sua aplicação será construída.

Eles ajudam no desenvolvimento rápido e seguro de aplicações.

### O que é Ionic?

Ionic é um framework para desenvolvimento de aplicações híbridas.

### Documentação do Ionic:

<https://ionicframework.com/docs>

## Instalação do NodeJS, Ionic e Visual Studio Code

### NodeJS

Basta entrar no site <https://nodejs.org/en/> e instalar a versão LTS normalmente.

### Ionic

Com o NodeJS instalado, basta digitar o seguinte comando no prompt do node:

```
> npm install -g ionic
```

### Visual Studio Code

Basta entrar no site <https://code.visualstudio.com/> e instalar normalmente.

## Construindo a aplicação

1 - Crie uma pasta chamada “projeto”.

2 - Abra a pasta “projeto” com o Visual Studio Code.

3 - Abra o terminal com o atalho `ctrl + `` (control + aspas simples).

4 - Gerando o projeto

```
> ionic start marvel
```

5 - Iniciando o projeto

```
> cd marvel
```

```
> ionic serve
```

### O que é o arquivo `app.module.ts`?

O Ionic é modular, ele precisa de módulos para gerenciar os nossos componentes, esse então é o módulo padrão.

### O que é o módulo `HttpClient`?

O `HttpClient` possibilita ao front-end comunicar com o back-end, sendo assim ele pode receber e mandar informações para o banco de dados.

# MARVEL

## 6 - Importando o modulo HttpClient no app.module.ts

```
import { HttpClientModule } from '@angular/common/http';
```

Obs: é necessário colocar nos imports depois do BrowserModule

```
@NgModule({  
  imports: [  
    BrowserModule,  
    HttpClientModule,  
  ],  
})
```

### O que é um service?

O objetivo principal de um serviço é organizar e compartilhar lógicas, modelos, informações e funções entre diferentes componentes de uma aplicação.

## 7 - Criando o service

> ionic generate service services/heros/heros

## 8 - Importando HttpClient no heros.service.ts

```
import { HttpClient } from '@angular/common/http';
```

### O que é um construtor?

O método construtor é utilizado para instanciar módulos em classes.

## 9 - Instanciando o HttpClient no construtor

```
constructor(private http: HttpClient) { }
```

### O que é o md5?

O md5 é um algoritmo de hash, ele é usado para criptografar informações.

### Por que a API usa md5?

A API da marvel, assim como muitas outras, utilizam md5 para segurança dos dados, o md5 é uma forma de verificar se o usuário tem permissões para utilizar rotas da mesma, caso o usuário não passe um token(md5) como parâmetro ou passe um token errado, a API não vai executar o método.

## 10 - Instalando md5

> npm install ts-md5

## 11 - Importando o modulo md5 no heros.service.ts

```
import { Md5 } from 'ts-md5/dist/md5';
```

### O que é timestamp?

Timestamp é uma cadeia de caracteres denotando a hora ou data que certo evento ocorreu.

### Por que a API usa timestamp?

A API usa timestamp para salvar quando as requisições são feitas, e gerar um token unico, sendo assim, um token gerado agora não será válido após alguns segundos.

# MARVEL

## 12 - Criando a função para trazer todos os heróis da Marvel

```
getAllHeros() {  
  const md5 = new Md5();  
  const timestamp = Number(new Date());  
  const hash = Md5.hashStr(timestamp +  
'ebd407c102ea3f1262b8dd370cfa04d4a132a867d8b23f3429d72898aaffd1a321761b4a');  
  
  return  
  this.http.get(`https://gateway.marvel.com:443/v1/public/characters?ts=${  
timestamp}&orderBy=name&limit=20&apikey=d8b23f3429d72898aaffd1a321761b4a  
&hash=${hash}`).toPromise();  
}
```

## 13 - Usando o heros.service no home.page.ts

```
import { HerosService } from '../services/heros/heros.service';
```

```
constructor( private herosService: HerosService ) {}
```

```
export class HomePage implements OnInit {  
  public heros: any;  
}
```

```
getAllHeros() {  
  let result: any;  
  result = this.herosService.getAllHeros();  
  this.heros = result.data.results;  
}
```

## 14 - Criando lista na home.page.html

```
<ion-header>
  <ion-toolbar color="danger">
    <ion-title text-center>
      Marvel Heros
    </ion-title>
  </ion-toolbar>
</ion-header>

<ion-content>
  <ion-list>
    <ion-item>
      <ion-avatar slot="start">
      </ion-avatar>
      <ion-label>
        Teste!
      </ion-label>
    </ion-item>
  </ion-list>
</ion-content>
```

## 15 - Populando herois na lista

```
<ion-item *ngFor="let hero of heros">
  <ion-avatar slot="start">
    
  </ion-avatar>
  <ion-label>
    {{hero.name}}
  </ion-label>
</ion-item>
```

## 14 - Criando a página de detalhes

> ionic generate page hero-description

### O que são rotas?

Rotas são maneiras de navegar por páginas em aplicações, como por exemplo:

`http://localhost:8100/home`

`http://localhost:8100/hero-description`

Rotas também podem ser usadas para passar pequenas informações, como por exemplo o id de um usuário que será mostrado na próxima página:

`http://localhost:8100/hero-description/105`

## 15 - Alterando o app-routing.module.ts para receber o id do heroi

Antes

```
const routes: Routes = [  
  { path: '', redirectTo: 'home', pathMatch: 'full' },  
  { path: 'home', loadChildren: './home/home.module#HomePageModule' },  
  { path: 'hero-description', loadChildren: './hero-description/hero-description.module#HeroDescriptionPageModule' },  
];
```

Depois

```
const routes: Routes = [  
  { path: '', redirectTo: 'home', pathMatch: 'full' },  
  { path: 'home', loadChildren: './home/home.module#HomePageModule' },  
  { path: 'hero-description/:id', loadChildren: './hero-description/hero-description.module#HeroDescriptionPageModule' },  
];
```

## 16 - Abrindo a página hero-description pela home.page.html

```
<ion-item [routerLink]="['/hero-description']" *ngFor="let hero of heros">
```

## 17 - Passando o id como parametro na rota

```
<ion-item [routerLink]="['/hero-description', hero.id]" *ngFor="let hero of heros">
```

## 18 - Pegando o id na página hero-description.page.ts

```
import { ActivatedRoute } from '@angular/router';
```

```
constructor(private route: ActivatedRoute) { }
```

```
ngOnInit() {  
    this.getDescription();  
}  
  
async getDescription() {  
    const heroId = await this.route.snapshot.params.id;  
    console.log(heroId);  
}
```

## 19 - Alterando heros.service.ts para passar id do heroi como parametro e receber detalhes do mesmo

```
getDescription(id) {  
    const md5 = new Md5();  
    const timestamp = Number(new Date());  
    const hash = Md5.hashStr(timestamp +  
        'ebd407c102ea3f1262b8dd370cfa04d4a132a867d8b23f3429d72898aaffd1a321761b4a');  
    return  
    this.http.get(`https://gateway.marvel.com:443/v1/public/characters/${id}  
        ?ts=${timestamp}&orderBy=name&limit=20&apikey=d8b23f3429d72898aaffd1a321761b4a&hash=${hash}`).toPromise();  
}
```

## 21 - Alterando hero-description.page.ts para passar id do heroi como parametro e receber detalhes

```
import { HerosService } from '../services/heros/heros.service';
```

```
export class HeroDescriptionPage implements OnInit {  
    public hero = {  
        name: null,  
        description: null,  
        thumb: null  
    };  
};
```

```
constructor(  
    private herosService: HerosService,  
    private route: ActivatedRoute  
) { }  
  
ngOnInit() {  
    this.getDescription();  
}  
  
async getDescription() {  
    const heroId = await this.route.snapshot.params.id;  
  
    if (heroId) {  
        let result: any;  
        result = await this.herosService.getDescription(heroId);  
        const data = await result.data.results;  
        this.hero.name = await data[0].name;  
        this.hero.description = await data[0].description;  
        this.hero.thumb = await  
`${data[0].thumbnail.path}.${data[0].thumbnail.extension}`;  
    }  
}
```

## 22 - Populando descrição do heroi na hero-description.page.html

```
<ion-content>

  <ion-card>

    

    <ion-card-content>

      <ion-card-title>

        {{hero.name}}

      </ion-card-title>

      <p>

        {{hero.description}}

      </p>

    </ion-card-content>

  </ion-card>

</ion-content>
```



# MARVEL

