

ANGULAR 2

Angular v2.2.1

[Home](#)

[Why Angular2?](#)

[Components](#)

[Inputs](#)

[Outputs](#)

[Lifecycle](#)

[Templates](#)

[Events](#)

[Forms](#)

[ViewChild](#)

ES6/TypeScript

VIEWCHILD

[Suggest improvements](#)

[Tweet](#)

Child components in our view can be accessed from our parent component easily with `@ViewChild`.

@ViewChild

To get access to a component and its methods, we can use the `@ViewChild` decorator.

For example, our `<user-profile>` component can have a method called `sendData()`.

CHILD.COMPONENT.TS

```
@Component({  
  selector: 'user-profile'  
})
```

[Wat?](#)

[Variables](#)

[Classes](#)

[Template Strings](#)

[Arrow Functions](#)

[Promises](#)

```
export class UserProfile {  
  constructor() {}  
  sendData() {  
    //send data  
  }  
}
```

When use the `user-profile` on our parent component, we can reference the `UserProfile` component class and then assign it to a local property:

PARENT.COMPONENT.TS

```
import { Component, ViewChild } from '@angular/core';  
import { UserProfile } from '../user-profile';  
  
@Component({  
  template: '<user-profile (click)="update()"></user-profile>',  
})  
export class MasterPage {
```

```
// ViewChild takes a class type or a reference name string
// Here we are using the type
@ViewChild(UserProfile) userProfile: UserProfile

constructor() { }

ngAfterViewInit() {
  // After the view is initialized, this.userProfile will be available
  this.update();
}

update() {
  this.userProfile.sendData();
}
}
```

We can also do the same thing with a local variable. Instead of trying to load the particular class, we can do:

PARENT.COMPONENT.TS

```
import { Component, ViewChild } from '@angular/core';
```

```
import { UserProfile } from '../user-profile';

@Component({
  template: '<user-profile #myProfile (click)="update()">'
})
export class MasterPage {
  @ViewChild('myProfile') userProfile: UserProfile
  constructor() { }
  update(){
    this.userProfile.sendData();
  }
}
```

Built by the [Ionic](#) Team. Licensed under Apache 2.