Simplifying ViewChild and ContentChild in Angular 10.

Reading Time: 5 minutes

Do you feel insecure, anxious and doubtful about ViewChild,View Children,ContentChild and ContentChildren decorators in Angular? No worries!!! I hope you will get clarity simply by reading this Blog.

Essentially, ViewChild, ViewChildren, ContentChild, and ContentChildren, are used for component communication in Angular. Therefore, if a parent component wants access to the child component then it uses these decorators.

@ViewChild()

If you want to access the following inside the parent component, use @ViewChild() decorator of Angular.

- DOM Element
- Directive
- Child Component

1. @ViewChild using DOM Element

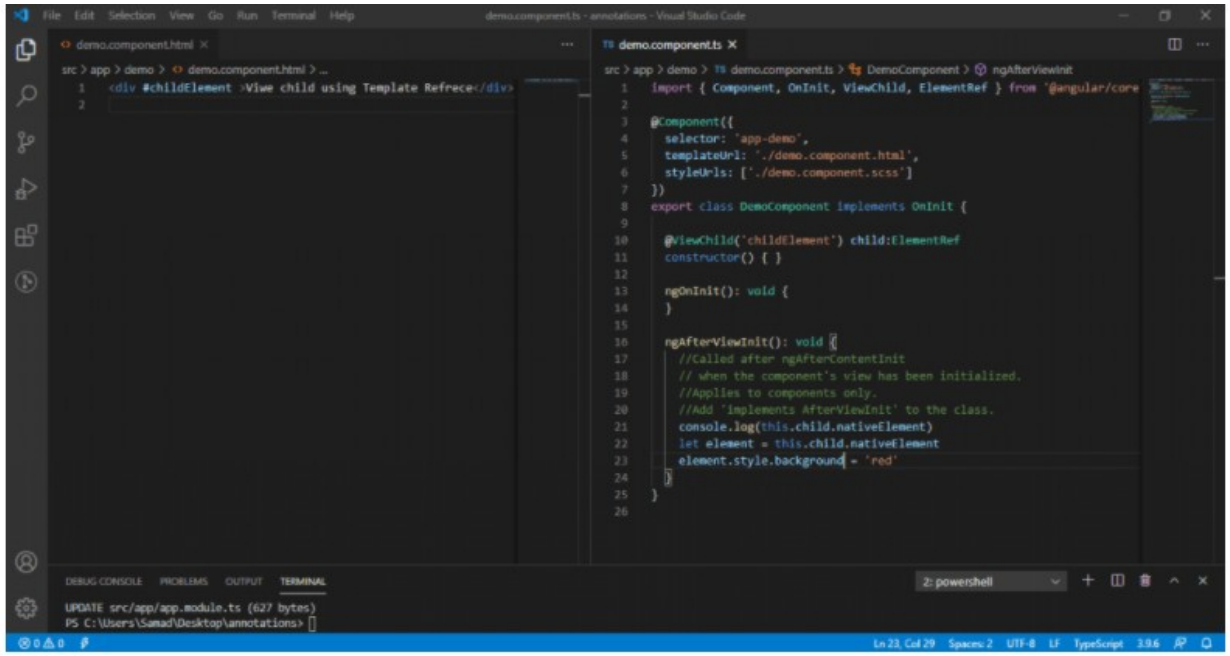
First create a component and In this component HTML add Template reference variable like <input #TemplateName> .Then in the controller file import dependencies @ViewChild(),ElementRef,AfterViewInit. Create an instance of this Template variable.Pass the template variable name in string as an argument to the @ViewChild() like below mentioned.

Samad
Author

Samad
Author
03 Dec, 2020

Simplifying ViewChild and ContentChild in Angular 10

Samad
Author



Here childElement is the template variable and in the controller file we are changing the color of the DOM element.

2. @ViewChild() using Directive:-

In this method, first, we need to create one directive.

```
DEBUG CONSOLE   PROBLEMS   OUTPUT   TERMINAL

PS C:\Users\Samad\Desktop\annotations> ng g d demo
CREATE src/app/demo.directive.spec.ts (216 bytes)
CREATE src/app/demo.directive.ts (137 bytes)
UPDATE src/app/app.module.ts (696 bytes)
PS C:\Users\Samad\Desktop\annotations>
```

In this directive, we need to import dependencies like ElementRef and AfterViewInit and create one method in this directive that changes the color of the text.

03 Dec, 2020

Simplifying ViewChild and ContentChild in Angular 10

Samad
Author
03 Dec, 2020

Simplifying ViewChild and ContentChild in Angular 10

```ts
import { Directive, ElementRef } from '@angular/core';

@Directive({
  selector: '[appDemo]'
})
export class DemoDirective {

  constructor(private eleRef:ElementRef) { }
  ngAfterViewInit(): void {
    //Called after ngAfterContentInit
    // when the component's view has been initialized. Applies to components only.
    //Add 'implements AfterViewInit' to the class.
    this.eleRef.nativeElement.style.color = 'red'
  }

  changeColor(color){
    this.eleRef.nativeElement.style.color = color
  }
}

```


Samad
Author
03 Dec, 2020

Simplifying ViewChild and ContentChild in Angular 10

Pass the directive selector into the parent component like this, Here appDemo is the directive name.

HTML:

```html
<div>Viwe child using Directive</div>
<p appDemo>change my color</p>
<div>
    <input type="radio" name="rd" (click)="changeDirectivedColor('green')">Green
    <input type="radio" name="rd" (click)="changeDirectivedColor('yellow')">Yellow
    <input type="radio" name="rd" (click)="changeDirectivedColor('blue')">Blue
</div>

```


Samad
Author
03 Dec, 2020

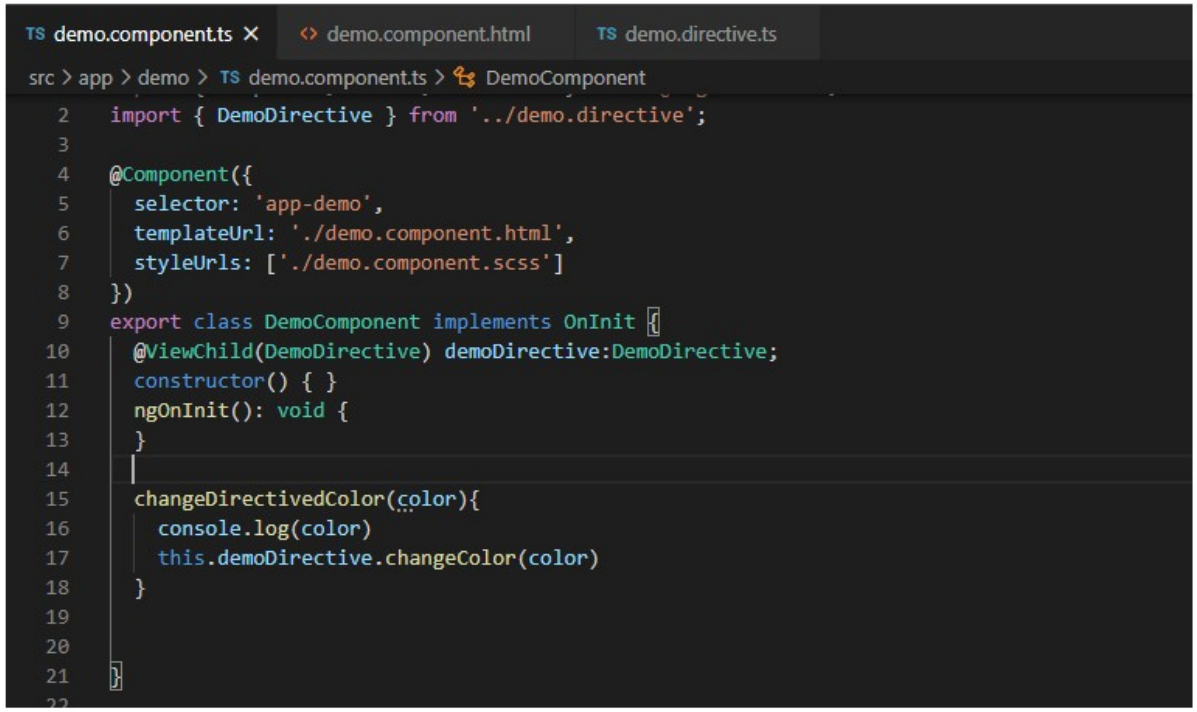
Simplifying ViewChild and ContentChild in Angular 10

Ts:

- In the parent component controller file, we need to import @ViewChild decorator and

pass the directive name as an argument to the @ViewChild() and then access the method which we created in the directive. In parent HTML we need to pass the directive selector as an attribute.

- Here demoDirective is an instance of DemoDirective and we are passing text color through changeDirectivedColor() method.



Samad
Author
03 Dec, 2020

Simplifying ViewChild and ContentChild in Angular 10

3. @ViewChild() using Child Component:

Create parent and child component in the below-mentioned format.

Parent component:

```
CREATE src/app/parent/parent.component.ts (276 bytes)
CREATE src/app/parent/parent.component.scss (0 bytes)
UPDATE src/app/app.module.ts (475 bytes)
PS C:\Users\Samad\Desktop\annotations>
PS C:\Users\Samad\Desktop\annotations>
```

Child component:

```
PS C:\Users\Samad\Desktop\annotations> ng g c child
CREATE src/app/child/child.component.html (20 bytes)
CREATE src/app/child/child.component.spec.ts (621 bytes)
CREATE src/app/child/child.component.ts (272 bytes)
CREATE src/app/child/child.component.scss (0 bytes)
UPDATE src/app/app.module.ts (553 bytes)
PS C:\Users\Samad\Desktop\annotations>
```

Create methods for increment and decrement in the child component to call in the parent component.

child.component.html

src > app > child > child.component.html > ...

```
1  <div>{{message}}</div>
2
```

child.component.ts

src > app > child > child.component.ts > ChildComponent > message

```
1  import { Component, OnInit } from '@angular/core';
2
3  @Component({
4    selector: 'app-child',
5    templateUrl: './child.component.html',
6    styleUrls: ['./child.component.scss']
7  })
8  export class ChildComponent implements OnInit {
9    number = 0;
10   message = `Counter :${this.number}`
11   constructor() { }
12
13   ngOnInit(): void {
14   }
15
16   increment(){
17     this.number = this.number+1;
18     this.message = `Counter :${this.number}`;
19   }
20   decrement(){
21     this.number = this.number-1;
22     this.message = `Counter :${this.number}`;
23   }
24 }
25
```

Import the child component and pass the child component class name as an argument to

Samad
Author
03 Dec, 2020

Simplifying ViewChild and ContentChild in Angular 10

Samad
Author
03 Dec, 2020

Simplifying ViewChild and ContentChild in Angular 10

the @ViewChild decorator and Call child component methods into parent component as mentioned in below.

HTML:

parent.component.html

```
src > app > parent > parent.component.html > div
<h1>ViewChild using child Component</h1>
<app-child></app-child>

<div>
    <button (click)="incrementChildCount()">Increment</button>
    <button (click)="decrementChildCount()">Decrement</button>
</div>
```

TS:

parent.component.ts | parent.component.html

```
src > app > parent > parent.component.ts > ParentComponent > constructor
import { Component, OnInit, QueryList, ViewChildren } from '@angular/core';
import { ChildComponent } from '../child/child.component';

@Component({
  selector: 'app-parent',
  templateUrl: './parent.component.html',
  styleUrls: ['./parent.component.scss']
})
export class ParentComponent implements OnInit {
  @ViewChildren(ChildComponent) childComponent:QueryList<ChildComponent>;
  constructor() { }
  ngOnInit(): void {

  }

  ngAfterViewInit(): void {
    //Called after ngAfterContentInit when the component's view has been initialized.Applies to components only.
    //Add 'implements AfterViewInit' to the class.
    console.log(this.childComponent.toArray())
  }

  incrementChildCount(id){
    this.childComponent.toArray()[id].increment()
  }
  decrementChildCount(id){
    this.childComponent.toArray()[id].decrement()
  }
}
```

Samad
Author
03 Dec, 2020

Simplifying ViewChild and ContentChild in Angular 10

Samad
Author
03 Dec, 2020

Simplifying ViewChild

@ViewChildren()

- If the parent component has more than one same child component then we are using @ViewChildren decorator.Use to get the QueryList of child Component from the view DOM.
- Any time a child element is added, removed, or moved, the query list will be updated, and the changes observable of the query list will emit a new value.View queries are set before the ngAfterViewInit callback is called.

Parent component:

HTML: In parent component two same child components are there.If we need to access both we need to use @ViewChildren decorator.

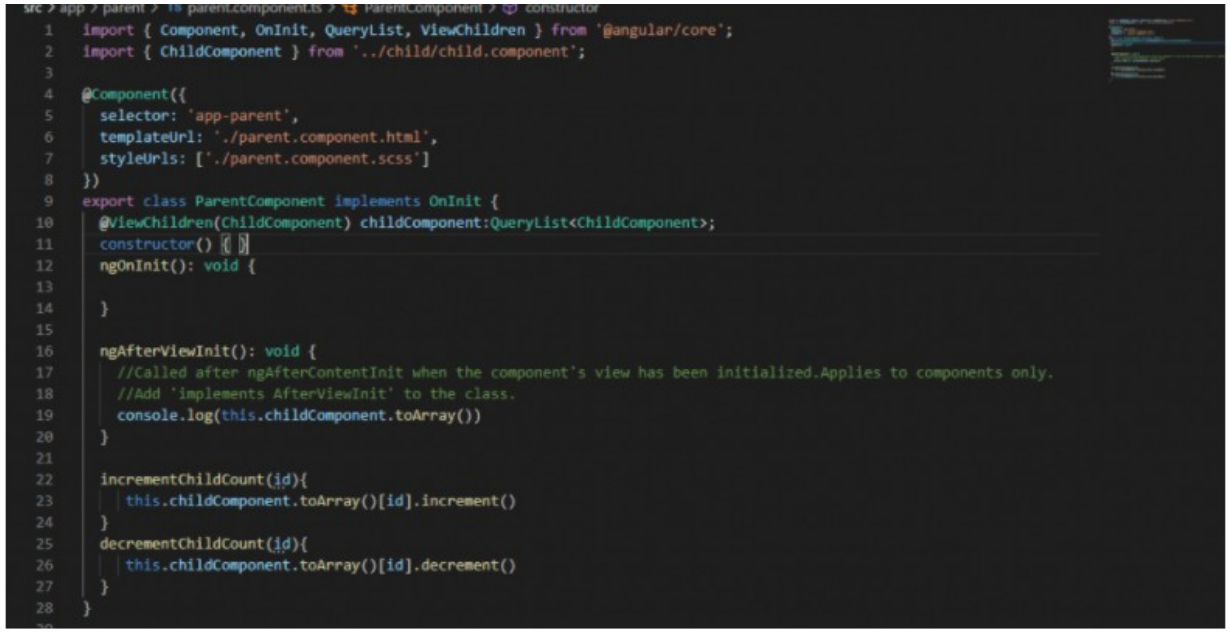
TS:

To access both child components in the parent component We need to convert the childComponent instance as an array like below mentioned this.childComponent.toArrary() Otherwise it will always trigger the first child component.

and ContentChild in Angular 10

Samad
Author
03 Dec, 2020

Simplifying ViewChild and ContentChild in Angular 10



```
import { Component, OnInit, QueryList, ViewChildren } from '@angular/core';
import { ChildComponent } from '../child/child.component';

@Component({
  selector: 'app-parent',
  templateUrl: './parent.component.html',
  styleUrls: ['./parent.component.scss']
})
export class ParentComponent implements OnInit {
  @ViewChildren(ChildComponent) childComponent:QueryList<ChildComponent>;
  constructor() { }
  ngOnInit(): void {

  }

  ngAfterViewInit(): void {
    //Called after ngAfterContentInit when the component's view has been initialized.Applies to components only.
    //Add 'implements AfterViewInit' to the class.
    console.log(this.childComponent.toArray())
  }

  incrementChildCount(id){
    this.childComponent.toArray()[id].increment()
  }
  decrementChildCount(id){
    this.childComponent.toArray()[id].decrement()
  }
}
```

Samad
Author
03 Dec, 2020

Simplifying ViewChild and ContentChild in Angular 10

@ContentChild():

Use to get the first element or the directive matching the selector from the content DOM. If the content DOM changes, and a new child matches the selector, the property will be updated.

Content queries are set before the ngAfterContentInit callback is called.

Does not retrieve elements or directives that are in other components' templates, since a component's template is always a black box to its ancestors.

We are creating student details by using child components and we are accessing or updating child component content using @ContentChild() decorator,@ContentChild always triggers first child component content.If we need to add content inside any component we need to use <ng-content> element.

Parent component:

Samad
Author
03 Dec, 2020

Simplifying ViewChild and ContentChild in Angular 10

HTML:

parent.component.html

```
src > app > parent > parent.component.html > ...
<h1>ContentChild useage</h1>
<ng-container *ngFor="let student of students">
    <app-child [student]='student'>
        <div>Subjects:
            <ng-container *ngFor="let subject of student.subjects">
                <app-subjects [subjects]='subject'></app-subjects>
            </ng-container>
        </div>
    </app-child>
</ng-container>
```

TS:

parent.component.ts

```
src > app > parent > parent.component.ts > ParentComponent > students
import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-parent',
  templateUrl: './parent.component.html',
  styleUrls: ['./parent.component.scss']
})
export class ParentComponent implements OnInit {
  subjects = [
    {
      e1:"Analog Electronics",
      e2:"Digital Electronics",
      e3:"VLSI"
    }
  ]
  students=[
    {
      "studentId": 1,
      "name": "Samad",
      "age": 22,
      "subjects":this.subjects
    },
    {
      "studentId": 2,
      "name": "Varma",
      "age": 21,
      "subjects":this.subjects
    },
    {
```

Samad
Author
03 Dec, 2020

Simplifying ViewChild and ContentChild in Angular 10

Samad
Author

```
    "studentId": 3,
    "name": "Mahendra",
    "age": 21,
    "subjects":this.subjects
  }
];

constructor() { }
ngOnInit(): void {}
}
```

Child component:

HTML:

```
ponent.html | parent.component.ts | child.component.html × | child.component.ts
src > app > child > child.component.html > ...
<div>
    <div>Name:{{student.name}}</div>
    <div>Age:{{student.age}}</div>
    <ng-content></ng-content>
</div>
```

TS:

```
ponent.html | parent.component.ts | child.component.html | child.component.ts × | subjects.co
src > app > child > child.component.ts > ChildComponent
import { SubjectsComponent } from '../subjects/subjects.component';

@Component({
  selector: 'app-child',
  templateUrl: './child.component.html',
  styleUrls: ['./child.component.scss']
})
export class ChildComponent implements OnInit {
  @ContentChild(SubjectsComponent) subjectsComponent:SubjectsComponent

  @Input() student;
  constructor() { }

  ngOnInit(): void {
```

03 Dec, 2020

Simplifying ViewChild and ContentChild in Angular 10

Samad
Author
03 Dec, 2020

Simplifying ViewChild and ContentChild in Angular 10

```ts
  }
  ngAfterContentInit(): void {
    //Called after ngOnInit
    //when the component's or directive's content has been initialized.
    //Add 'implements AfterContentInit' to the class.
    this.subjectsComponent.subjects.e1 = "Communications"
  }

}

```

Sub child component:

HTML:

```
ponent.html | parent.component.ts | child.component.html | child.
src > app > subjects > subjects.component.html > ...
<div>
    <div>Engineering 1:{{subjects.e1}}</div>
    <div>Engineering 2{{subjects.e2}}</div>
    <div>Engineering 3{{subjects.e3}}</div>
</div>
```

TS:

```
ponent.html | parent.component.ts | child.component.html | child.component.ts
src > app > subjects > subjects.component.ts > SubjectsComponent > ngOnInit
import { Component, OnInit, Input } from '@angular/core';

@Component({
  selector: 'app-subjects',
  templateUrl: './subjects.component.html',
  styleUrls: ['./subjects.component.scss']
})
export class SubjectsComponent implements OnInit {
  @Input() subjects
  constructor() { }

  ngOnInit(): void {
  }
```


Samad
Author
03 Dec, 2020

Simplifying ViewChild and ContentChild in Angular 10

Samad
Author
03 Dec, 2020

Simplifying ViewChild and ContentChild in Angular 10

```
15   }
16
```

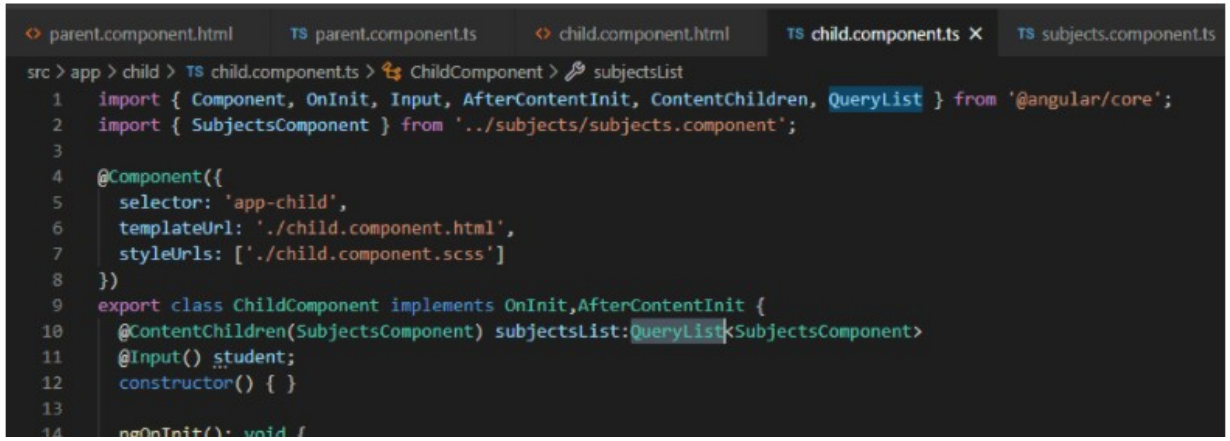
@ContentChildren():-

- To access or update all child components content we are using @ContentChildren decorator.
- Use to get the QueryList of elements or directives from the content DOM. Any time a child element is added, removed, or moved, the query list will be updated, and the changes observable of the query list will emit a new value.
- Content queries are set before the ngAfterContentInit callback is called.
- Does not retrieve elements or directives that are in other components' templates, since a component's template is always a black box to its ancestors.

Just we need to change the child component like this importing @ContentChildrent,QueryList.

Child component:

TS:



Samad
Author
03 Dec, 2020

Simplifying ViewChild and ContentChild in Angular 10

```
  }
  ngAfterContentInit(): void {
    //Called after ngOnInit
    //when the component's or directive's content has been initialized.
    //Add 'implements AfterContentInit' to the class.
      // this.subject.subjects.e1 = "Hello"
      this.subjectsList.toArray().forEach((component)=>{
        component.subjects.e1 = "Digital communication"
      })
  }

}
```

Samad
Author
03 Dec, 2020

Simplifying ViewChild and ContentChild in Angular 10

Well,

In this blog, we discussed @ViewChild,@ViewChildren,@ContentChild, and @ContentChildren decorators in Angular. These are excellent utilities for querying child elements in views. For a parent to child component communication, these are very useful and interesting decorators. Still, if you have any questions please feel free to ask questions in the comment box and I will write another blog answering those. Thank you!!!

Share

Leave a Comment

Name

Email

Leave a Comment

Recent Articles

Tips for B2B Design and Development

- Sarath

03 Dec, 2020

Accessibility in design: How to get started?

- Anurag

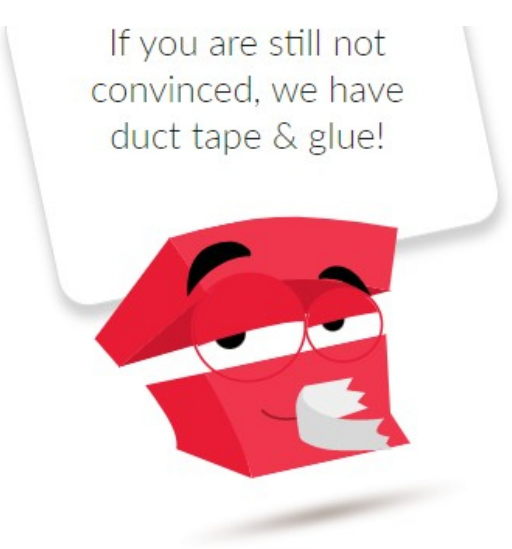
03 Dec, 2020

Key Benefits of IoT in Agriculture & Smart Farming

- Sushree

03 Dec, 2020

We can also make your ideas sticky. What's your idea?



Name

Email

Company Location (Country)

Mobile Number (Optional)

Let's Connect ⟶

India
Vasista Bhavan, 3rd Floor,
A.P.H.B Colony, Gachibowli,
Hyderabad, Telangana.

India
10th floor, RMZ Latitude Commercial,
Bellary Rd, Hebbal,
Bengaluru.

TOP UX AGENCIES Clutch 2020

GoodFirms TOP APP DESIGNING COMPANY goodfirms.co

USA
Divami Inc 951 Mariners Island Blvd,
Suite 300 PMB3005,
San Mateo, CA 94404.

Phone
+91 (40) 6733 7033
+1 (408) 634 8266

★★★★★ 43 REVIEWS

★★★★★ 53 REVIEWS

Email

connect@divami.com
hr@divami.com