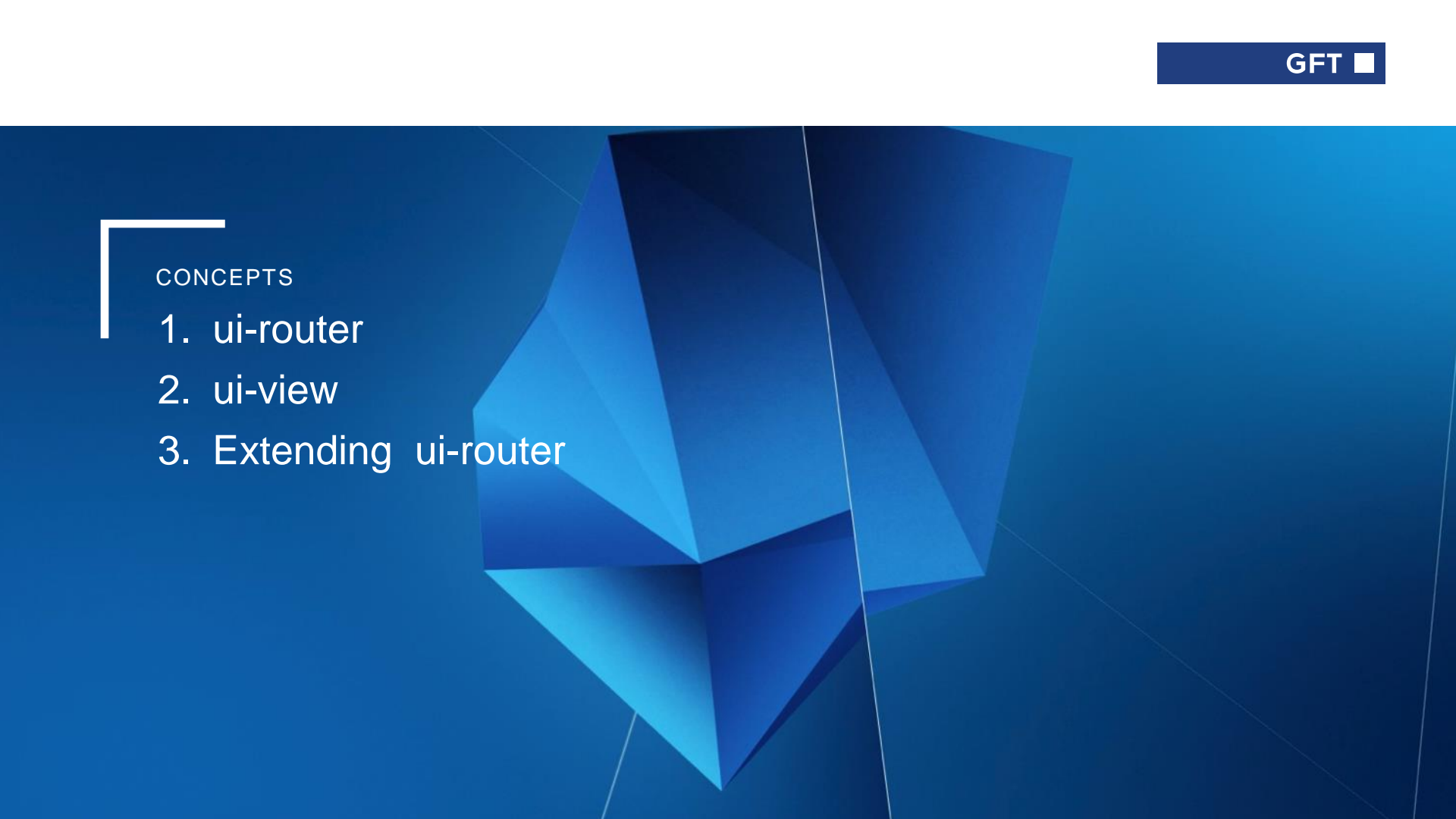UI-ROUTER , STATES AND VIEW

Routing , States and Views

CONCEPTS

1. ui-router
2. ui-view
3. Extending ui-router

GFT ■

- The UI-Router is a routing framework for AngularJS. It provides a different approach than ngRoute in that it changes your application views based on state of the application and not just the route URL.

App states file

```
…
$stateProvider
// Define State object
    .state('example', {
        // Url from the state
        url: '/Example',
        // Template url
        templateUrl: 'components/exampleComponent/example.html',
        // Controller definiton for the state
        controller: 'GeneratorController'
    })
// Creating a nested state extending the previous one
    .state('example.details', {
        // Url will be www.yourapp.com/example/details
        url: '/details',
        templateUrl: 'components/exampleComponent/example.html',
        controller: 'GeneratorController'
    });
…
```

- ▪ Is a directive that extend the regular routing behaviour creating containers having the ability to nest states & views.

<div style="text-align:center">html</div>

```html
<!-- index.html (root unnamed template) -->
<body ng-app>
<div ui-view ui-view="header" ></div> <!-- header.html plugs in here -->
<div ui-view="body"></div><!-- body.html plugs in here -->
</body>
```

NOTE:
To navigate between state ui-router use a different tag replacing the href by ui-sref="NameOfState"

<div style="text-align:center">App states</div>

```javascript
…
$stateProvider

// Define State object
  .state('example', {
    // Url from the state
    url: '/Example',
    //Block of nested views
    views: {
      'header': {
        templateUrl: 'components/exampleComponent/example-header.html'
      controller: function ($scope) { ... controller stuff just  header  view ... }
    },
      'body': {
       templateUrl: 'components/exampleComponent/example-body.html'
      controller: 'bodyController'
    },
    }
    })
…
```

- You can use resolve to provide your controller with content or data that is custom to the state. resolve is an optional map of dependencies which should be injected into the controller.

Controller

```
angular
    .module("App.Controllers")
    .controller("bodyController" , bodyController);

    function($scope , sampleData){
      // SampleData provided from the resolve on the defined state
      $scope.data = sampleData;
    }
```

App states

```
$stateProvider

// Define State object
  .state('example', {
    // Url from the state
    url: '/Example',
    templateUrl: 'components/exampleComponent/example-body.html'
    controller: 'bodyController' ,
    resolve:{
      sampleData:  function(){
        return {value: 'Data to Pass to our controller'};
      },
    }
  })
…
```

onEnter / onExit

- Trigger the execution of code on the events onEnter or onExit from the state.

Controller

```
angular
    .module("App.Controllers")
    .controller("bodyController" , bodyController);

    function($scope , sampleData){
      // SampleData provided from the resolve on the defined state
      $scope.data = sampleData;
    }
```

App states

```
$stateProvider.state("contacts", {
    template: '<h1>{{title}}</h1>',
    resolve: { title: 'My Contacts' },
    controller: function ($scope, title) {
        $scope.title = 'My Contacts';
    },
    onEnter: function (title) {
        if (title) { ... do something ... }
    },
    onExit: function(title) {
        if (title) { ... do something ... }
}
})
```