Scope Events Propagation IN ANGULARJS

- AngularJS provides an effective way to exchange messages to scopes at different hierarchical level.
- AngularJS provides **$emit** and **$broadcast** functions to achieve the **event propagation** in a hierarchical manner.

$emit Function

- The **$emit** function is used to propagate events upwards through the scope hierarchy.
- The event life cycle starts at the scope on which **"$emit"** was called.
- Thereafter, the event traverses upwards towards the **root scope** and calls every registered listener along the way.
- If one of the user cancels the event, then the **$emit** will stop propagating.

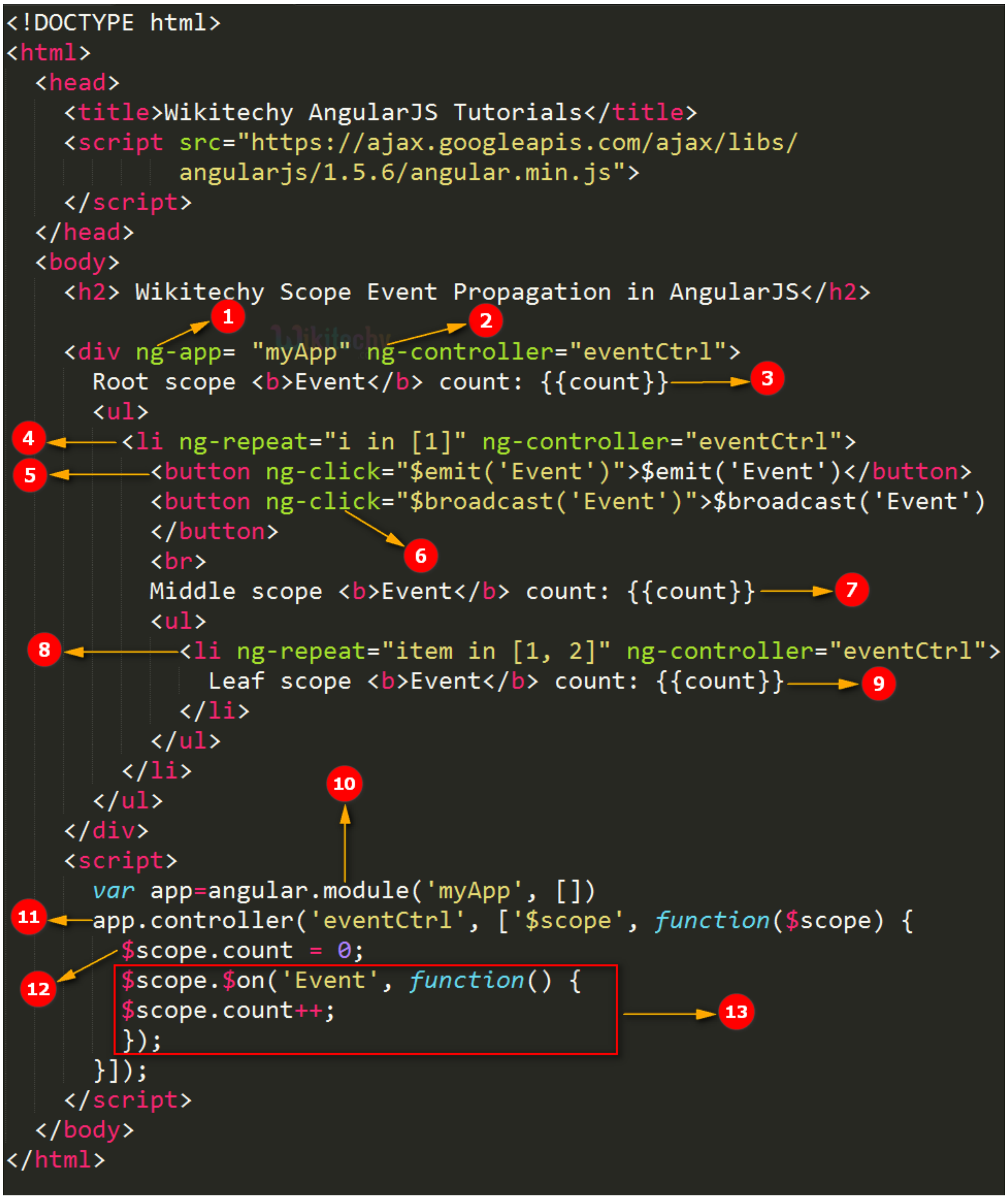
$broadcast Function

- The **$broadcast** function is used to propagate events downwards to every child scopes and their children scopes.
- The event life cycle starts at the scope on which **"$broadcast"** was called.
- All **listeners** listening for event on this scope get notified.
- Thereafter, the event traverses downwards towards the child scopes and calls every registered listener along the way.
- The $broadcast event can't be **canceled.**

Sample coding for Scope Event Propagation in AngularJS:

```
<!DOCTYPE html>
<html>
        <head>
                <title>Wikitechy AngularJS Tutorials</title>
        </head>
                <script src="https://ajax.googleapis.com/ajax/libs/
                        angularjs/1.5.6/angular.min.js"></script>
        <body>
               <h2> Wikitechy Scope Event Propagation in AngularJS</h2>
               <div ng-app= "myApp" ng-controller="eventCtrl">
                 Root scope <b>Event</b> count: {{count}}
                 <ul>
                    <li ng-repeat="i in [1]" ng-controller="eventCtrl">
                      <button ng-click="$emit('Event')">$emit('Event')</button>
                      <button ng-click="$broadcast('Event')">$broadcast('Event')
                      </button><br>
                      Middle scope <b>Event</b> count: {{ count }}
                     <ul>
                      <li ng-repeat="item in [1, 2]" ng-controller="eventCtrl">
                         Leaf scope <b>Event</b> count: {{ count }}
                      </li>
                     </ul>
                   </li>
               </ul>
           </div>
           <script>
              var app=angular.module('myApp', [])
              app.controller('eventCtrl', ['$scope', function($scope) {
                $scope.count = 0;
                $scope.$on('Event', function() {$scope.count++; });
              }]);
           </script>
        </body>
</html>
```

Code Explanation for Scope Event Propagation in AngularJS:



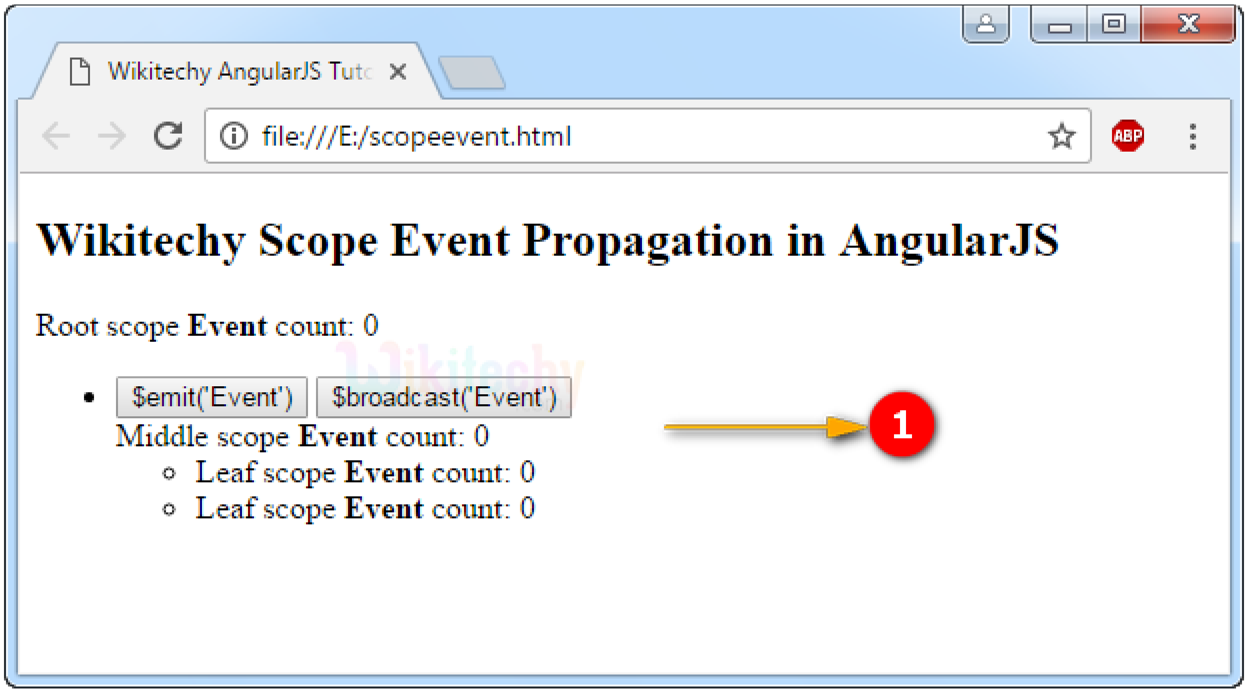
```
<!DOCTYPE html>
<html>
  <head>
    <title>Wikitechy AngularJS Tutorials</title>
    <script src="https://ajax.googleapis.com/ajax/libs/
            angularjs/1.5.6/angular.min.js">
    </script>
  </head>
  <body>
    <h2> Wikitechy Scope Event Propagation in AngularJS</h2>

    <div ng-app= "myApp" ng-controller="eventCtrl">
      Root scope <b>Event</b> count: {{count}}
      <ul>
        <li ng-repeat="i in [1]" ng-controller="eventCtrl">
          <button ng-click="$emit('Event')">$emit('Event')</button>
          <button ng-click="$broadcast('Event')">$broadcast('Event')
          </button>
          <br>
          Middle scope <b>Event</b> count: {{count}}
          <ul>
            <li ng-repeat="item in [1, 2]" ng-controller="eventCtrl">
              Leaf scope <b>Event</b> count: {{count}}
            </li>
          </ul>
        </li>
      </ul>
    </div>
    <script>
      var app=angular.module('myApp', [])
      app.controller('eventCtrl', ['$scope', function($scope) {
        $scope.count = 0;
        $scope.$on('Event', function() {
        $scope.count++;
        });
      }]);
    </script>
  </body>
</html>
```

1. The AngularJS application is defined by ng-app**="myApp".** The application runs inside the <div> tag. It's also used to define a <div> tag as a root element.
2. The ng-controller**=" eventCtrl"** is an AngularJS directive. It is used to define a controller name as **"eventCtrl".**
3. The **{{count}}** is used to bind the root Scope event count when the user click the **$emit** event button which is defined in the **eventCtrl** in JavaScript.
4. The ng-repeat is an AngularJS directive. It is used to repeat an item for root Scope and middle scope.
5. **<button ng-click="$emit('Event')">** is used to create a button and it is used to invoke the **$emit(Event)** when the button was clicked.
6. **<button ng-click="$broadcast('Event')">** is used to create a button and it is used to invoke the **$broadcast('Event')** when the button was clicked.
7. Here the **{{count}}** is used to bind the Middle Scope event count when the user click both the **$emit event** and **$broadcast event** button.
8. The ng-repeat is an AngularJS directive. It is used to repeat an item for middle scope and leaf scope.
9. Here the **{{count}}** is used to bind the Leaf Scope event count when the user click a **$broadcast event** button.

10. angular.module function is used to create a module. Here we have passed an empty array to it.
11. Here we have declared a controller module using **.controller()** function. The value of the controller modules is stored in scope object. In AngularJS, **$scope and $rootScope** are passed as first argument to **.controller()** during its constructor definition.
12. Here we have set the value of **$scope.count as "0"**(zero).
13. An event raised by **$broadcast()** and **$emit()** can be handled by wiring an event handler using **$on()** function. Here the value of count will be increased when the user click the $broadcast() and $emit() event.

Sample Output for Scope Event Propagation in AngularJS:

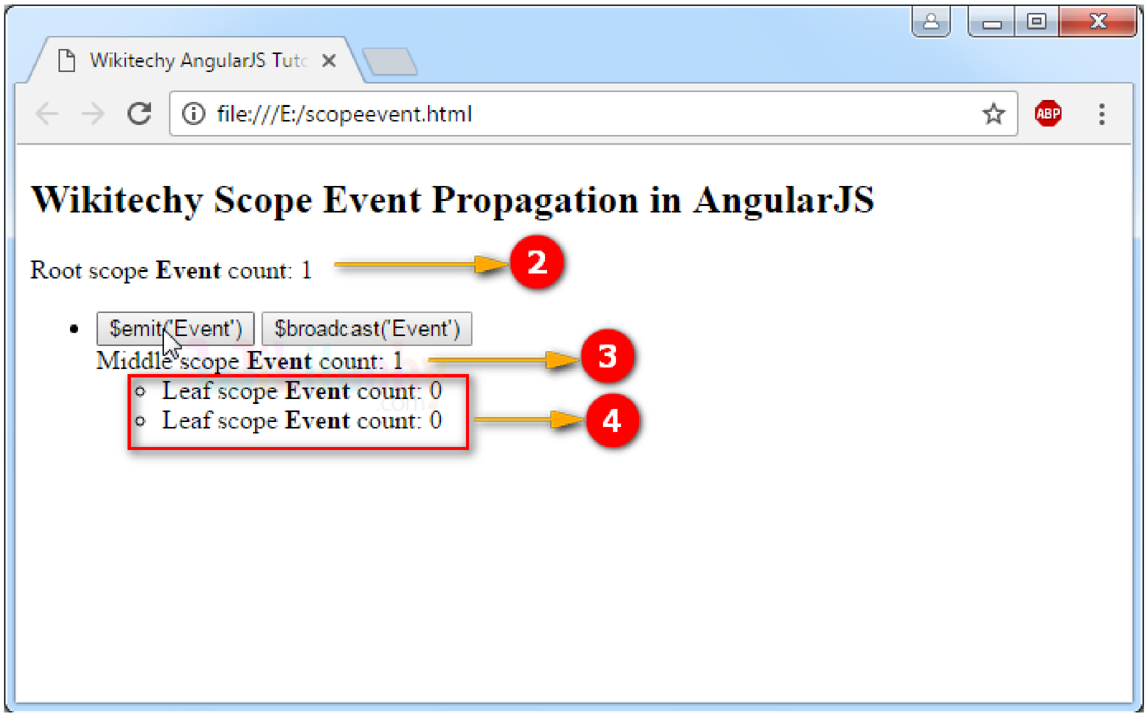


Wikitechy Scope Event Propagation in AngularJS

Root scope **Event** count: 0

- Middle scope **Event** count: 0
  - Leaf scope **Event** count: 0
  - Leaf scope **Event** count: 0

1. The page loaded with two button and content.

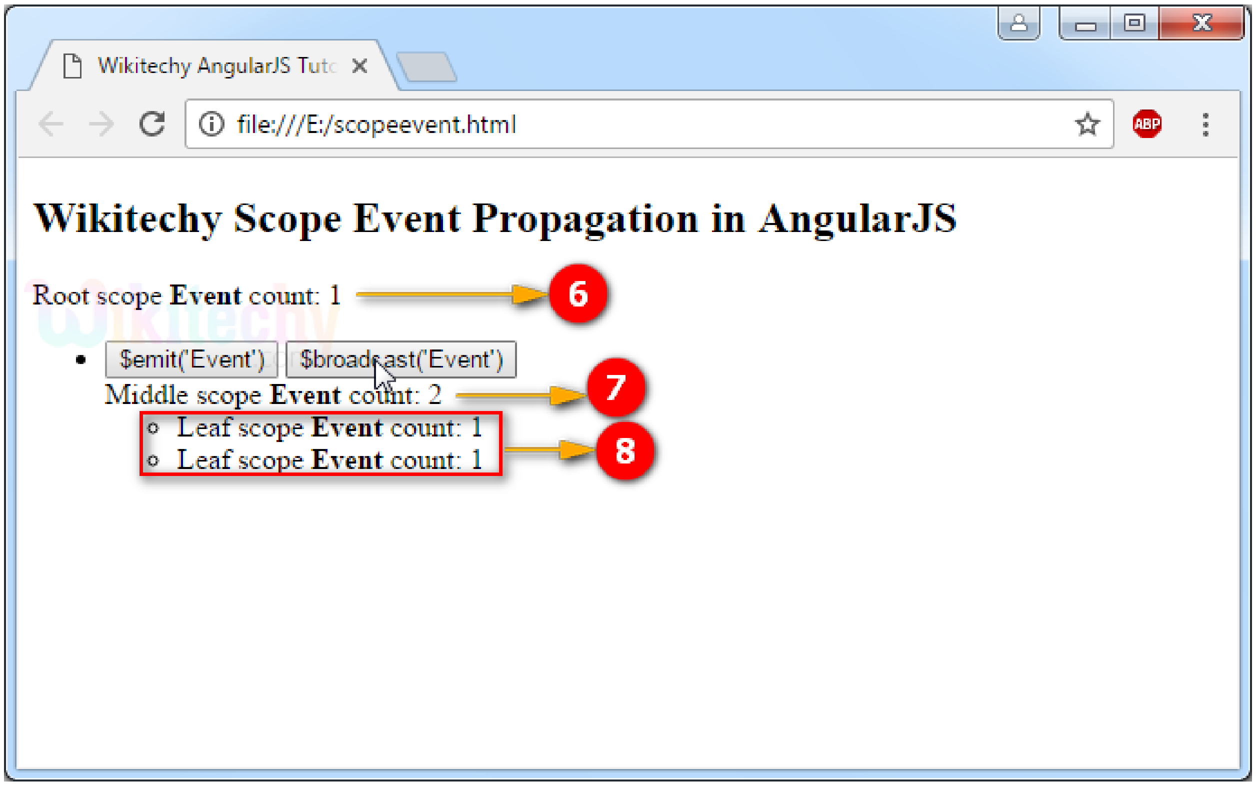


Wikitechy Scope Event Propagation in AngularJS

Root scope **Event** count: 1

- $emit('Event') $broadcast('Event')
  Middle scope **Event** count: 1
  - Leaf scope **Event** count: 0
  - Leaf scope **Event** count: 0

2. When the user click the **$emit('Event')** button the **"Root scope Event count"** is increased by **1.**
3. When the user click the **$emit('Event')** button the **"Middle scope Event count"** is increased by **1**.
4. When the user click the **$emit('Event')** button the **"Leaf scope Event count"** is does not increased because the **$emit()** function is only traverse through the parent Scope.



Wikitechy Scope Event Propagation in AngularJS

Root scope **Event** count: 1

- $emit('Event') $broadcast('Event')
  Middle scope **Event** count: 2
  - Leaf scope **Event** count: 1
  - Leaf scope **Event** count: 1

5. When the user click the **$broadcast('Event')** button the root scope event count is not increased the value because, the **$broadcast()** function is only traverse through the child scope.
6. When the user click the **$broadcast('Event')** button the middle scope event count is increased by **1**, now the value of middle scope event count is **2.**
7. When the user click the **$broadcast('Event')** button the Leaf Scope Event count is increased by **1,** now the value is **1.**