Understanding $watch(), $watchgroup() and $watchCollection() methods of scope

## **$watch**

This function is used to observe changes in a variable on the $scope. It accepts three parameters: expression, listener and equality object, where listener and equality object are optional parameters.

1. **$watch(watchExpression, listener, [objectEquality])**

Here, watchExpression is the expression in the scope to watch. This expression is called on every $digest() and returns the value that is being watched.

The listener defines a function that is called when the value of the watchExpression changes to a new value. If the watchExpression is not changed then listener will not be called.

The objectEquality is a boolean type which is used for comparing the objects for equality using angular.equals instead of comparing for reference equality.

1. **<script>**
2. **scope.name = 'User';**
3. **scope.counter = 0;**
5. **scope.$watch('name', function (newVal, oldVal) {**
6. **scope.counter = scope.counter + 1;**
7. **});**
8. **</script>**

## **$watchgroup**

This function is introduced in Angular1.3. This works the same as $watch() function except that the first parameter is an array of expressions to watch.

1. **$watchGroup(watchExpression, listener)**

The listener is passed as an array with the new and old values for the watched variables. The listener is called whenever any expression in the watchExpressions array changes.

1. **<script>**
2. **$scope.teamScore = 0;**
3. **$scope.time = 0;**
4. **$scope.$watchGroup(['teamScore', 'time'], function(newVal, oldVal) {**
5. **if(newVal[0] > 20){**
6. **$scope.matchStatus = 'win';**
7. **}**
8. **else if (newVal[1] > 60){**
9. **$scope.matchStatus = 'times up';**
10. **});**
11. **</script>**

## **$watchCollection**

This function is used to watch the properties of an object and fires whenever any of the properties change. It takes anobject as the first parameter and watches the properties of the object.

1. **$watchCollection(obj, listener)**

The listener is called whenever anything within the obj has been changed.

1. **<script>**
2. **$scope.names = ['User', 'deepak', 'mohit', 'kapil'];**
3. **$scope.dataCount = 4;**
5. **$scope.$watchCollection('names', function (newVal, oldVal) {**
6. **$scope.dataCount = newVal.length;**
7. **});**
8. **</script>**