

Search catalog

X Lessons

Lesson 1 - Components and Component-Based

Architecture

Lesson 2 - AngularJS Event System and Modules

Lecture 34, Part 1:
AngularJS Event System

8 min

Lecture 34, Part 2: AngularJS Event System

14 min

Practice Quiz:

Quiz 33 questions

Lecture 35, Part 1: Modules

7 min

Lecture 35, Part 2: Modules

10 min

3 questions

Practice Quiz:
Quiz 34

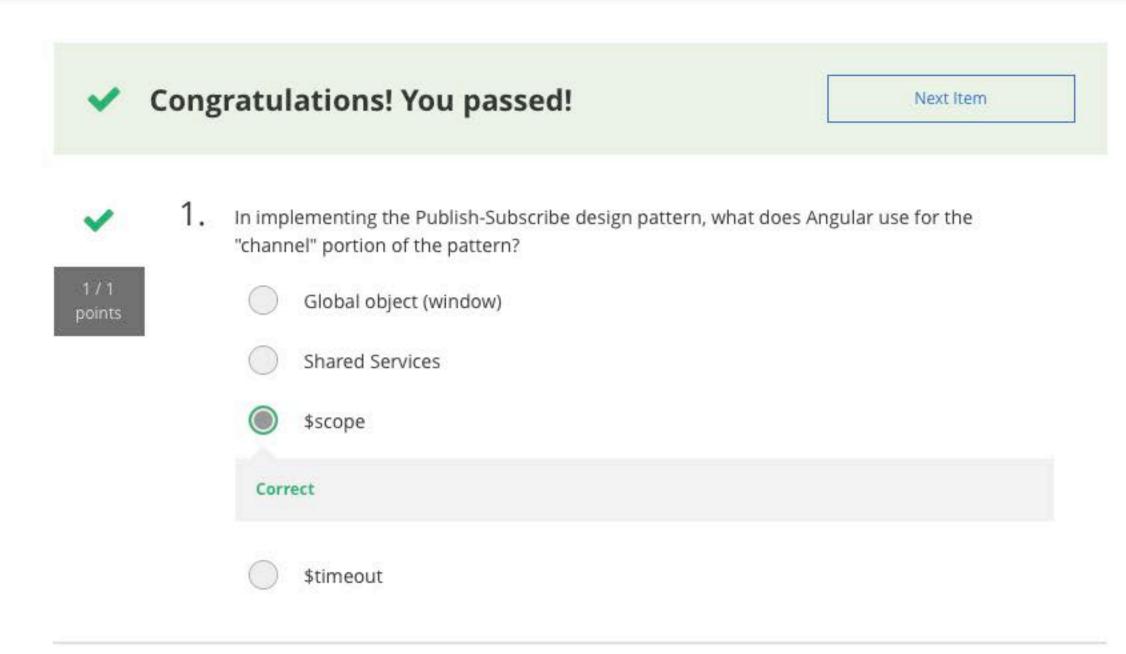
Lesson 3 -Introduction to Single Page Routing With uirouter

Lesson 4 - Handling Data with Routing



Practice Quiz, 3 questions

3/3 points (100%)





2. If your component is set up to listen for some event using the \$rootScope, what must you ensure to do in such a setup? For example, what must be done (if ANYTHING!) when you see the following in your component:

1 / 1 points



Quiz 33
Practice Quiz, 3 questions

3/3 points (100%)

- Aside from implementing the handler function, not much left to do. Angular takes care of the rest.
- The handler function must call the \$emit method with the event name of 'alert:turnOff'.
- Weeeeeee! Ok, clearly not the right answer, but so much fun! 😀
- We must deregister the handler when this view is destroyed. We can do this by declaring a local to controller variable called 'cancelHandler' and changing this line 2 to this:

```
1 cancelHandler = $rootScope.$on('alert:turnOn', handler);
```

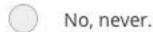
Then, add another lifecycle method to the component like this:

```
1 - $ctrl.$onDestroy = function () {
2   cancelHandler();
3 };
```

Correct



Is it ever OK to call \$rootScope.\$emit(...)?



111



Quiz 33 Practice Quiz, 3 questions

3/3 points (100%)

z cancernanater();

3 };

Correct



Is it ever OK to call \$rootScope.\$emit(...)?

1/1 points

No, never.



Yes, calling \$emit on the \$rootScope would make ONLY the \$rootScope in the application have a chance at catching that event.

Correct

While this is true, if you're finding yourself doing something like that, it probably means your architecture is messed up... USUALLY, you shouldn't be forced into such communications.





