



AngularJS - Promises

Egghead (/egghead)

```
<div>
  <ng-view></ng-view>
</div>
```

```
var app = angular.module('app', []);

app.config(function ($routeProvider) {
  $routeProvider
    .when('/', {
      templateUrl: "app.html",
      controller: "AppCtrl"
    })
  });

app.controller("AppCtrl", function ($scope) {
  $scope.model = {
    message: "This is my app!!!"
  }
});
```

```
<h1>{{ model.message }}</h1>
```

Angular ships with the promise library, \$q. \$q's defer() method returns a defer object.

```
app.controller("AppCtrl", function ($scope, $q) {

  var defer = $q.defer();

  defer.promise.then(function () {
    alert("I promised I would show up");
  });

  defer.resolve();

  $scope.model = {
    message: "This is my app!!!"
  }
});
```

Here, the promise.then() prepares the function passed to it to execute at some point in the future. The execution of the promise is achieved by invoking resolve().

Promises can also be chained:

```
defer.promise
  .then(function () {
    alert("I promised I would show up");
  })
  .then(function () {
    alert("me too");
  })
  .then(function () {
    alert("and I");
  });

defer.resolve();
```

This simply queues them in order to the same resolve() call.

It is also possible to resolve with parameters. The chained promises will cascade their return values to the subsequent promise:

```
defer.promise
  .then(function (weapon) {
    alert("You can have my " + weapon);
    return "bow";
  })
  .then(function (weapon) {
    alert("And my " + weapon);
    return "axe";
  })
  .then(function () {
    alert("And my " + weapon);
  });

defer.resolve("sword");
```

Obviously, this alerts "You can have my sword", "And my bow", "And my axe" in order.

You're finished!

🐦 [Share on Twitter \(https://twitter.com/intent/tweet?original_referer=https%3A%2F%2Fthinkster.io&text=Just%20finished%20AngularJS%20-%20Promises%20tutorial\)](https://twitter.com/intent/tweet?original_referer=https%3A%2F%2Fthinkster.io&text=Just%20finished%20AngularJS%20-%20Promises%20tutorial)

f [Share on Facebook \(https://www.facebook.com/sharer/sharer.php?u=https%3A%2F%2Fthinkster.io%2Fegghead%2Fpromises%2F\)](https://www.facebook.com/sharer/sharer.php?u=https%3A%2F%2Fthinkster.io%2Fegghead%2Fpromises%2F)