

Custom Services



Controller's Responsibilities

✧ Use controllers to

- Set up initial state of \$scope
- Add behavior to the \$scope



✧ Do **NOT** use controllers to

- ...
- Handle business logic directly
- Share code or state across controllers



Register Service Function Constructor

```
angular.module('app', [])  
  .controller('ctrl', Ctrl)  
  .service('CustomService', CustomService);
```

Use this name to inject
it into other services,
controllers, etc.

Treated as a
Function Constructor



Singleton Design Pattern

Restricts object to always having a single instance

- ✧ Each dependent component gets a reference to the same instance
- ✧ Multiple controllers injected with a Service will all have access to the same service instance



Lazily Instantiated

Only created if an application component declares it as a dependency

- ✧ If no components in your application are dependent on this service, it will never get created



Summary

- ✧ Controllers are not supposed to
 - Handle business logic
 - Code sharing
 - Be used to share data across other components
- ✧ Custom services instantiated with `.service` method
 - Singletons (only 1 instance of object exists)
 - Lazily instantiated (only created if something depends on them)
- ✧ `.service('name', function)`, treats function as a function constructor