

# Unit Testing RequireJS Modules

Jeff Valore  
CodingWithSpike.wordpress.com  
@CodingWithSpike



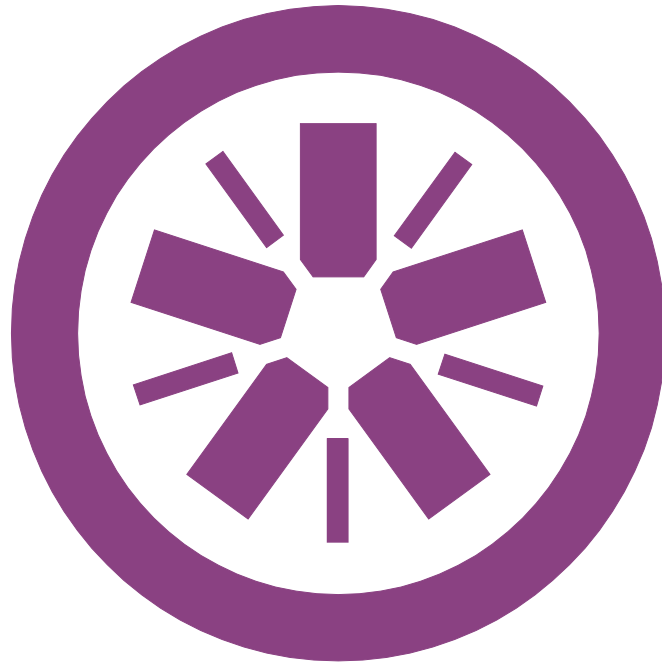
**pluralsight**   
hardcore dev and IT training

# Advantages of **Modules** and **RequireJS**

- Smaller, more **maintainable** pieces of code.
- **Single responsibility** in each module.
- **Dependency injection** allows mocking.
- Code is more **testable**.

## Difficulties of **Testing** with **RequireJS**

- Module loading is **asynchronous**.
- RequireJS **caches** modules between tests.



# Jasmine

<http://jasmine.github.io/>

# Jasmine Function Syntax

- Group tests together into test suites using the **describe** function.

```
describe("{test suite title}", {test suite function});
```

- Create a test with the **it** function.

```
it("{test title}", {test function});
```

# Matchers Follow an expect()

- `.toBe(...)` Matches with `===`
- `.toEqual(...)` Literal and Object equality
- `.toMatch(...)` Matches with a regular expression
- `.toBeDefined()` Compares against undefined
- `.toBeNull()` Compares against null
- `.toBeTruthy()` Matches boolean truthy
- `.toBeFalsy()` Matches boolean falsy
- `.toContain(...)` Finds an item in an array
- `.toBeCloseTo(...)` Matches that a value is within a range
- `.toThrow()` Tests that a function throws an exception
- `.toBeLessThan(...)` Matches with `<`
- `.toBeGreaterThan(...)` Matches with `>`



<https://github.com/iammerrick/Squire.js/>

# Summary

- **Jasmine test framework supports asynchronous testing.**
  - Call the “done” function when finished testing.
- **RequireJS can load modules into separate contexts.**
  - Create contexts by passing “context” to require.config()
- **Squire.js can manage RequireJS contexts and mock modules for us.**