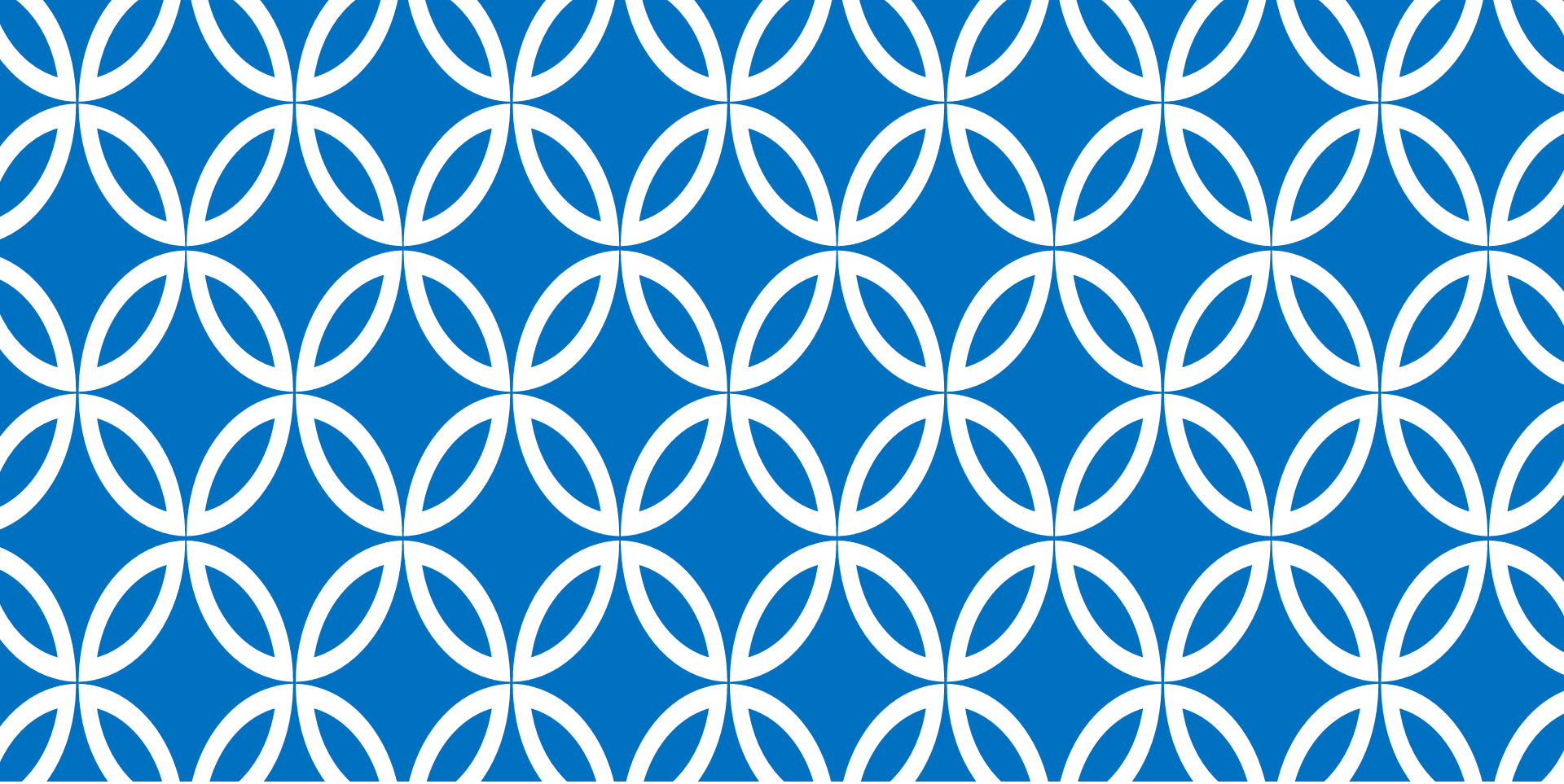
LEARN. DO. EARN

ACADGILD

FRONT END DEVELOPMENT (WITH ANGULARJS)

Website : http://www.acadgild.com
LinkedIn : https://www.linkedin.com/company/acadgild
Facebook : https://www.facebook.com/acadgild



Session 18 – AngularJS Testing

Agenda – AngularJS Testing

1. Automated Testing
2. List Of Frameworks Required For Testing Jasmine
3. Jasmine
4. Jasmine Spec
5. What Is A Suite?
6. Setup And Tear Down For Tests
7. Setting Up And Creating Tests
8. Setting Up Unit Testing For AngularJs

Automated Testing

- **Unit Testing:**
  - Unit testing is for testing the code.
  - It's similar to the way we test server side application..i.e. JUnit.
  - For unit testing JavaScript applications we can use **Jasmine.**

- **End to End testing:**
  - Sometimes unit testing is not enough.
  - Then we need to know how the application behaves to the user interactions in the browser.
  - For this we need to simulate user interactions at the browser.
  - For End to End testing we can use **Protractor.**

- **Note:** We will be using protractor testing framework built over Selenium and Jasmine.

List Of Frameworks Required For Testing Jasmine

- **Karma:** Karma is a task runner for unit testing. It assist in running the test and provided a lot more features to make testing easier.

- **Protractor:** Protractor is the End to end testing framework. It is used to write and execute e2e tests for angular.

Jasmine

- **Jasmine** is a behavior-driven development unit testing framework for testing JavaScript code.
- **Jasmine can be downloaded** from http://jasmine.github.io
- Jasmine can also be installed using npm(node package manager).
- Use the command in
  npm install -g jasmine
- Alternatively you can download the standalone version of Jasmine.

Jasmine Spec

- An **expectation** in Jasmine is an assertion that is either true or false.
- A **spec** is a collection of exceptions
- A spec with all expectations as true is a **passing spec**.
- If a spec has one or more false expectations, it is a **failing spec**.

Example of a spec:

```
it("and has a positive case", function() { var a = 5,b=5;
expect(a).toBe(b); });
```

What Is A Suite?

- A suite is a collection of specs.

- **Example of a suite:**

```
describe("A suite", function() {
it("and has a positive case", function() { var a = 5,b=5;
expect(a).toBe(b); }); });
```

Setup And Tear Down For Tests

- Following are global functions:
- The **beforeEach** function is called once before each spec in the describe is run.

- The **afterEach** function is called once after each spec.

- The **beforeAll** function is called only once before all the specs in a describe are run.

- The **afterAll** function is called after all specs finish.

Setting Up And Creating Tests

- Download the standalone version of jasmine.

- After downloading we have the following folders
  - **lib:** The lib folder contains the libraries for jasmine
  - **spec:** spec is the folder where will be keeping our tests
  - **src:** the folder where our code to be tested will be kept

- We also have a file called specrunner.html that we run within browser to run our tests.

Setting Up Unit Testing For AngularJS

- To start with we need to install **karma** and **karma cli** on the system.
- This can be done either using npm(node package manager) or you just need to run **npm install - g karma** and **npm install -g karma-cli**.
- The above two commands install karma on the machine.
- Once done we can start setting up our environment.

Agenda – AngularJS Testing

1. Setting Up protractor For End To End Testing
2. Running The Test

Setting Up Protractor For End To End Testing

- Install protractor using the command **npm install -g protractor**.

- After that update the web driver by **npm install webdriver-manager update**.

Running The Test

- To run the test we need to start the selenium server.
- Next use the command **webdriver-manager start** to start the server.
- After the server is started, we will open a new terminal.
- Go to **testing/project_e2e/** directory.
- Then we will fire the command **protractor config.js**
  This command will load the config.js in the protractor and start the testing.
- Once chrome browser get opened, values will get filled automatically and then tested by the framework.

Agenda – AngularJS Testing

1. Introduction To Karma
2. Generate karma.config.js

Introduction To Karma

- **Karma** is a test runner developed by the AngularJS team.
- It's basically the engine that goes through all the unit tests, runs them and then provides the results of the executed unit tests.
- Once Karma is installed, you will need a karma.config.js file so that Karma knows where the JavaScript files are located, what testing framework you want to use, etc.

Generate karma.config.js

- **You can generate a karma.config.js** file by running the following command:

  karma init

- **When you run this you will be asked a bunch of questions:**

  1. Which unit testing framework do you use?
  2. Do you want to use Require.js?
  3. Do you want to capture any browsers automatically?
  4. What is the location of your source and test files?
  5. Should any of the files included by the previous patterns be excluded?
  6. Do you want karma to watch all the files and run the tests on change?

Accept all the defaults. If needed, go back and change the configuration later.

Lets Discuss Assignments