Dojo Development

Course Summary

Description

This three-day course introduces students to a large part of the Dojo library. There is a strong focus on the Dijit UI library as well, including general Dijit concepts, a tour of Dijit capabilities and two sections on how to build your own Dijit widgets. After completing this course, students should be able to understand and build front-end, web-based UIs via Dojo and Dijit.

Topics

- Intro to Dojo
- Basic Dojo: DOM and Events
- Querying the DOM
- Dojo and Events
- The AMD Pattern and Dojo
- Dojo and Classes
- Dojo and Remote Data Access
- Testing with QUnit
- Storing data
- Introduction to Dijit
- Dijit Widgets
- Creating widgets through subclassing
- Creating your own widgets
- Dojo's Grid: dgrid
- Final Project: Details to be provided by Mathworks

Audience

Engineers and programmers who will be building front-end user interfaces using the Dojo framework. Not intended for new programmers, and not intended for those new to JavaScript.

Prerequisites

Students should be comfortable with programming in general, preferably in a modern language like Java, C, or C#. Students should be comfortable with JavaScript, including JavaScript types, functions, objects, and syntax.

Duration

Three days

Dojo Development

Course Outline

I. Intro to Dojo
- A. What Dojo is and isn't
- B. Our first, basic Dojo script
- C. How to do things the Dojo way
- D. Asynchronous Modules and AMD
- E. Requiring the code you need
- F. Loading Dojo
- G. Dojo and the nano core
- H. dojoConfig: Configuring how Dojo is loaded

II. Basic Dojo: DOM and Events
- A. dojo/dom: Asking for an element by name
- B. What do I get back? DOMNodes and Nodes
- C. dojo/on: Event handling with Dojo
- D. Basic event handling
- E. The event handling function
- F. The event object
- G. Back to the DOM
- H. Creating DOM elements
- I. Accessing DOM properties and attributes
- J. Manipulating CSS (dojo/dom-style and dojo/dom-class)

III. Querying the DOM
- A. dojo/query queries elements by CSS selector
- B. Overview of CSS selectors
- C. Processing NodeLists
- D. Additional NodeList features
- E. The DOM interface
- F. Events with NodeLists
- G. Traversal
- H. Effects

IV. Dojo and Events
- A. Brief reminder of dojo/on
- B. Removing event handlers
- C. Specific event types
- D. Unloading the window
- E. Mouse Events
- F. Keyboard events
- G. Touch events
- H. Event delegation

V. The AMD Pattern and Dojo
- A. Creating standalone modules with define()
- B. Returning objects, functions and classes
- C. Configuring libraries
- D. Using require() to control what is loaded and when it is loaded

VI. Dojo and Classes
- A. dojo/declare(): Creating classes
- B. Defining methods and properties
- C. Inheritance and Mixins
- D. Constructors

VII. Dojo and Remote Data Access
- A. Introduction
- B. Overview of remote data concepts
- C. Brief introduction of Mathworks API
- D. All remote data access works with promises
- E. Promises and Deferreds
- F. What's a promise?
- G. What's a Deferred?
- H. Reacting to a Deferred: then
- I. Chaining Deferreds
- J. Using when to handle promises and non-promises
- K. Dojo's list of promises

VIII. Testing with QUnit
- A. Introduction to JavaScript unit testing
- B. Introduction to QUnit
- C. Qunit syntax
- D. Organizing tests
- E. Asnychronous testing
- F. DOM testing
- G. Automated testing with QUnit and karma

IX. Storing data
- A. The new school: dojo/store
- B. The dojo/store API
- C. dojo/store is more of an interface than an implementation
- D. Mathworks' data store objects

X. Introduction to Dijit
- A. What is Dijit?
- B. Declarative vs Programmatic
- C. In this course, we will be using exclusively programmatic code
- D. Dijit concepts
- E. Dijit tools
- F. Common Dijit methods and properties
- G. dojo/Stateful

Dojo Development

Course Outline (cont'd)

XI. Dijit Widgets
- A. Form widgets
- B. Layout widgets
- C. Other widgets

XII. Creating widgets through subclassing
- A. Understanding the inheritance chain
- B. The Dijit lifecycle
- C. A tour of Dijit mixins
- D. Subclassing the right Dijit

XIII. Creating your own widgets
- A. Working from scratch with _WidgetBase
- B. Custom events with dojo/Evented
- C. Managing state with dojo/Stateful
- D. Building from a template with _TemplatedMixin
- E. Widgets within widgets: _WidgetsInTemplateMixin

XIV. Dojo's Grid: dgrid
- A. Acquiring dgrid
- B. dgrid philosophy
- C. A basic dgrid
- D. Enhancing with mixins and extensions
- E. Tying a dgrid to a store
- F. A dynamically updating dgrid

XV. Final Project: Details to be provided by Mathworks