AureliaJS

Jesse Javier Cogollo Alvarez

Developer by passion

email: cogollo87@gmail.com

MedellinJS

August 18, 2015

Contenido

AureliaJS

DEMO

Que es Aurelia?

Aurelia es un Framework javascript al lado del servidor. Que maneja simples convenciones. http://aurelia.io/

Rob Eisenberg robeisenberg.com

Caracteristicas

1. **Forward-thinking**

Escrito con ES6 y ES7. se integra con Web components. no tiene dependencias externas (excepto polyfills).

Caracteristicas

1. **Forward-thinking**
2. **Modern Architecture**

Aurelia esta compuesto por pequeños módulos. Utilizados juntos como un completo Framework. Pero también permite personalizarlo con diferentes módulos.

Caracteristicas

1. **Forward-thinking**
2. **Modern Architecture**
3. **Two-Way Databinding**

Aurelia tiene una poderosa two-way binding. el cual utiliza técnicas adaptativos para utilizar el mecanismo mas eficiente para observar cada propiedad. Object.observer

Caracteristicas

1. **Forward-thinking**
2. **Modern Architecture**
3. **Two-Way Databinding**
4. **Extensible HTML**

La extensibilidad HTML de Aurelia permite crear elementos HTML personalizados, añadir nuevos comportamientos para elementos ya existentes.

Caracteristicas

1. **Forward-thinking**
2. **Modern Architecture**
3. **Two-Way Databinding**
4. **Extensible HTML**
5. **Routing and UI Composition**

Router con acoplamiento activo, patrones de ruta dinámica, activación de pantallas asincrónicamente, etc.

Caracteristicas

1. **Forward-thinking**
2. **Modern Architecture**
3. **Two-Way Databinding**
4. **Extensible HTML**
5. **Routing and UI Composition**
6. **MV* with Conventions**

Quieres invertir tiempo en escribir código de configuración? Aurelia maneja una convención simple.

Caracteristicas

1. **Forward-thinking**
2. **Modern Architecture**
3. **Two-Way Databinding**
4. **Extensible HTML**
5. **Routing and UI Composition**
6. **MV* with Conventions**
7. **Broad Language Support**

Utiliza ES5, ES6, TypeScript, AtScript, CoffeeScript.

Caracteristicas

1. **Forward-thinking**
2. **Modern Architecture**
3. **Two-Way Databinding**
4. **Extensible HTML**
5. **Routing and UI Composition**
6. **MV* with Conventions**
7. **Broad Language Support**
8. **Testable**

combinando módulos ES6 con contenedores inyección de dependencias. se hace mas fácil crear código altamente cohesivo y bajo acoplamiento. haciendo las pruebas unitarias algo fácil.

Como empezar - Estructura

Descargamos la ultima versión del skeleton de Aurelia.

https://github.com/aurelia/skeleton-navigation/releases/tag/0.15.1

descomprimimos y ya tenemos una estructura lista para utilizar:

- build (configuracion gulp).
- doc (documentacion).
- src (codificacion).
- styles (estilos CSS).
- test (pruebas).
- README.md (instrucciones para empezar con AureliaJS).

Modulos

- **logging**

Esta librería es parte de la plataforma Aurelia y contiene un "Appender" para el logging.

Implementación: **welcome.js**

Modulos

- **Plugins**

Aurelia permite crear e implementar plugins que se adaptan al Framework. implementación:

index.html

animation-main.js

Vistas y vistas modelos

- **DI inyeccion de dependencias**

Los AureliaElements son creados como clases los cuales son llamados por el Framework usando contenedor de inyección de dependencias.

implementación: **flickr.js**

Vistas y vistas modelos

- **Plantillas**

El motor de plantillas de Aurelia es el encargado de cargar las vistas y sus recursos, compilando su HTML para un optimo desempeño y renderizado del UI en la pantalla.

<template></template>. Todo lo que este adentro de esas etiquetas será administrado por Aurelia.

implementación: **welcome.html**

Databiding

- **Que es**

El Databinding nos permite enlazar el estado y comportamiento en un objeto(js) y una vista(html).

Cualquier cambio es enlazado y/o sincronizado en una o varias direcciones. Se puede diferenciar en la vista por que el elemento va seguido de un punto en el atributo con el valor (bind, one-way, two-way o one-time).

implementación: **welcome.html**

Bindings

- **Delegate, trigger and call**

Los bindings no solamente conectan elementos o atributos. También son utilizados para lanzar comportamientos. Se recomienda utilizar el delegate por defecto por que es el mas eficiente al utilizar la memoria y la CPU.

Implementación: **TODO**

Bindings

- **String interpolation**

Sintaxis: ${nameVariable}. es un binding de un solo camino, la cual la salida se convierte en un string.

Implementación: **welcome.html**

Bindings

- **Ref**

Es especial para conectar elementos personalizados, utilizando esta técnica puedes conectar diferentes componentes.

Implementación: **TODO**

Bindings

- **Select elements**

"value.bind" es un HTMLSelectElement que tiene un comportamiento especial para soportar seleccion simple y selección múltiple de valores. normalmente se combina con el elemento repeat. Pero existe otro elemento especial que permite trabajar con objetos "model.bind".

Implementación: **TODO**

Bindings

- **Radios**

"checked.bind" es un HTMLInputElement tiene un comportamiento especial para soportar binding no booleano, como strings y objetos.

Implementación: **TODO**

Bindings

- **Checkboxes**

Implementación: **TODO**

Bindings

- **InnerHTML**

podemos hacer binding de los elementos innerHTMl con el atributo innerhtml.

Implementación: **TODO**

Bindings

- **textContent**

Podemos hacer binding de los elementos textContect con el atributo textcontent.

Implementación: **TODO**

Bindings

- **Style**

Podemos hacer binding a los string CSS u objetos en un elemento style.

Implementación: **TODO**

Bindings

- **Adaptive binding**

Implementación: **TODO**
Important

HTML Extensions

- **show**

Nos permite condicionar si un elemento o elementos serán visibles o no en el DOM.
Implementación: **TODO**

HTML Extensions

- **if**

Tiene el mismo comportamiento que el "show". lo que lo hace diferente es que esta etiqueta si remueve el componente del DOM.

Implementación: **TODO Important**

HTML Extensions

- **repeat**

Este atributo permite renderizar un template múltiples veces y trabaja en conjunto con el ".for"

Implementación: **TODO**
Important

HTML Extensions

- **repeat $parent**

Permite acceder a las propiedades de la viewModel.

Implementación: **TODO Important**

HTML Extensions

- **repeat $index**

Es el Index del ítem en el array.

Implementación: **TODO**
Important

HTML Extensions

- **repeat $first**

Es verdadero si es el primer ítem del array.

Implementación: **TODO Important**

HTML Extensions

- **repeat $last**

Es verdadero si es el último ítem del array.

Implementación: **TODO Important**

HTML Extensions

- **repeat $even**

Es verdadero si el ítem es un número par.

Implementación: **TODO Important**

HTML Extensions

- **repeat $odd**

Es verdadero si el ítem es un número impar.

Implementación: **TODO Important**

Routing

- **Descripción**

Cuando implementas enrutadores, tienes varias posibilidades de implementarlo, como: navegación de la App, dashboards y MDI interfaces.

El Routing de Aurelia, el Router que vive en los viewModels y los router-view que viven en las vistas.

Routing

- **Que podemos hacer**

rutas estaticas: **/contactUs**

rutas parametriables:
/user/:userId/dasboard

rutas wilcard: **file*path**

Routing

- **Ciclo de vida**

  Existen 4 estados:

canActivate: permite validar si se puede navegar en el viewModel.

activate: permite implementar lógica antes del que el viewModel sea cargado.

canDeactivate: permite controlar si deseo que en ese momento se pueda abandonar el viewModel.

deactivate: permite implementar lógica una vez el viewModel ya fue cambiado.

Routing

- **Convencíon de enrutadores**

Implementación: **TODO**
Important

Routing

- **Personalizando el enrutador**

Aurelia nos permite agregar pasos en el proceso del Router, los que nos permite agregar validaciones de permisos.

Implementación: **TODO Important!!!**

Routing

- **pushState**

si así lo prefieres, puedes remover el hash de la URL.

Implementación: **TODO Important!!!**

Routing

- **Reutilizando un viewModel existente**

Si alguna vez usted quiere implementar la misma vista para diferentes rutas.

Implementación: **TODO Important**

Routing

- **Renderizando multiples viewPorts**

Si alguna vez necesitas renderizar contenido en mas de un área con la misma ruta.

Implementación: **TODO Important !!!**

Routing

- **Rutas personalizadas**

Aurelia nos permite generar Routes personalizados.

Implementación: **TODO Important!!!**

Extendiendo HTML

- **Que es**

Aurelia tiene un compilador de templates. Existen dos tipos de extensiones:

Atributos personalizados

Elementos personalizados

Extendiendo HTML

- **Atributos personalizados**

Los atributos personalizados extienden el HTML con funcionalidades añadiendo nuevos comportamientos para elementos existentes.

Implementación: **TODO Important**

Extendiendo HTML

- **Elementos personalizados**

Los elementos personalizados crean nuevos tags al HTML.

Implementación: **say-hello.js (TODO)**

Eventos

- **Que es**

Es una útil herramienta para cuando se necesita desacoplar componentes y necesita hablar con otro componente.

Eventos

- **Eventos DOM**

Debe ser utilizado cuando un mensaje especifico es enviado.

https://developer.mozilla.org/en-US/docs/Web/Guide/Events/Creating_and_triggering_events

HTTP Client

- **Que es**

Aurelia incluye su propia interface para comunicarse con el objeto XMLHttpRequest.

Tiene la mayoría de los verbos estándares. además incluye jsonp. Cuando realizas una petición, este retorna una promesa con el objeto HttpResponseMessage.

Personalización (Customization)

- **Que es**

Implementación: **TODO**
Important

DEMO

=)

Redes sociales

1. **Twitter**

@jessecogollo

Redes sociales

1. Twitter
2. **Facebook**

/jessecogollo

Redes sociales

1. Twitter
2. Facebook
3. **Github.com/jessecogollo**

https://github.com/jessecogollo/ama

Donde aprender

1. **Pagina web oficial** http://aurelia.io/get-started.html

Donde aprender

1. Pagina web oficial
2. **pluralsight**

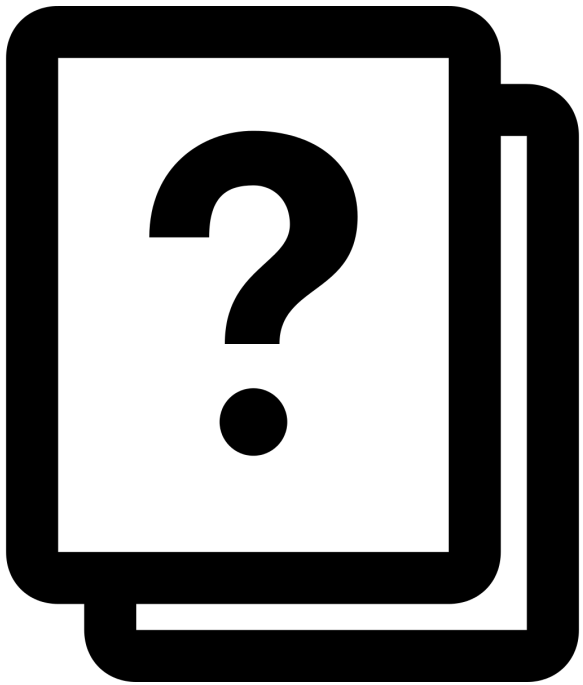
http://www.pluralsight.com/courses/building-applications-aurelia

Donde aprender

1. Pagina web oficial
2. pluralsight
3. **youtube**

https://www.youtube.com/results?search_query=aurelia+js

Preguntas



Gracias !!! =)