aurelia
HTML
CSS
JavaScript
TypeScript
webpack
MODULE BUNDLER
node
JS
Sass

Arjen de Blok

- Senior Technical Consultant bij ICT Groep (www.ict.eu) sinds 1995
- Programmeren sinds 1990
- Technologiën
  - Visual C++ met Microsoft Foundation Classes
  - .NET WinForms & WPF
  - Silverlight
  - ASP.NET WebForms, MVC, Web API
  - Microsoft SharePoint
  - KnockoutJS, PagerJS, TypeScript, SASS
  - AngularJS, TypeScript, SASS
  - Aurelia, WebPack, TypeScript, SASS
- Links
  - Github: https://github.com/arjendeblok

Agenda

1. Basics for every web developer
2. Road to the Client Side
3. Putting it all together
4. Aurelia
5. Demo RecipeWeb

   Pauze
6. Back to the Server Side with ASP.NET Core
7. Testing your Aurelia App

Part 1

Basics for every web developer

Web Development Everywhere

- For the browser
  - Web Sites
  - Web Apps
- For mobile
  - Apache Cordova
- For desktop
  - Windows 10 UWP
  - Electron
- For smart watch
  - Samsung Tizen

HTML 5

- Previous version 4.01 released in 1999!
- Draft since 2007
- Finalized in 2014
- Internet Explorer support 9+
- New document type
- Consists of many features → Check for browser support!
  - http://html5test.com/
- Works in conjuction with CSS 3 and JavaScript

Semantic HTML

- HTML5 tags
  - <header>, <footer>
  - <article>
  - <nav>
  - <aside>
- List tags
  - , <ol>,
- Use them in CSS
  - Less class selectors

HTML 5 Additional Features

- New Input Types
  - Number, Date, Time, Range
- Graphics
  - Vector - SVG
  - Bitmap - Canvas
  - 3D - WebGL
- Video & Audio
  - Native Tag
  - Streaming
- Storage
- Web Sockets

CSS 3

- Extension of CSS 2
- Additional CSS Selectors like attribute and child selectors.
- Border styles like border-radius and box-shadow.
- Text effects like text-shadow
- 2D/3D Transformations
- Animations
- Transitions

JavaScript

- Specification: ECMAScript
- Version ES5 supported on all browsers
- ECMAScript 2015 or ES6 partially supported on Edge, Chrome, FireFox
- ECMAScript 2016 or ES7 partially supported on Edge, Chrome, FireFox
- Transpilers for compiling ES 2015/2016 to ES5
  - Babel
  - TypeScript 2.1
  - Traceur
  - https://kangax.github.io/compat-table/es6/
- Important: ES6 Promise pattern.

DEMO

TypeScript

- Superset of JavaScript
- Transpiles to JavaScript (Transpiler); ES5 or ES2015
  - Not all can be transpiled to ES5!
- Can make variables ‘typed’ => Type checking and intellisense.
- Type ‘any’ also exists for unknown types
- Integrated with Visual Studio (2015 or Code)
- Does not make JavaScript an OO language!
- Definitions from
  - DefinitelyTyped
  - Typings
  - NPM repository

DEMO

Responsive Web Design

- Past
  - Web site has a fixed with of 960px with spacing left and right.
  - Designer provides PhotoShop picture of 960px wide.
  - Seperate site for mobile devices.
- Modern
  - Signle web sites that adapts layouts to resolution of device
  - Responsive Web Design.
  - More interaction Designer and Developer.
- Frameworks
  - ZURB Foundation
  - Twitter Bootstrap
  - MaterializeCSS

Part 2

Road to the Client Side

Web Development Options

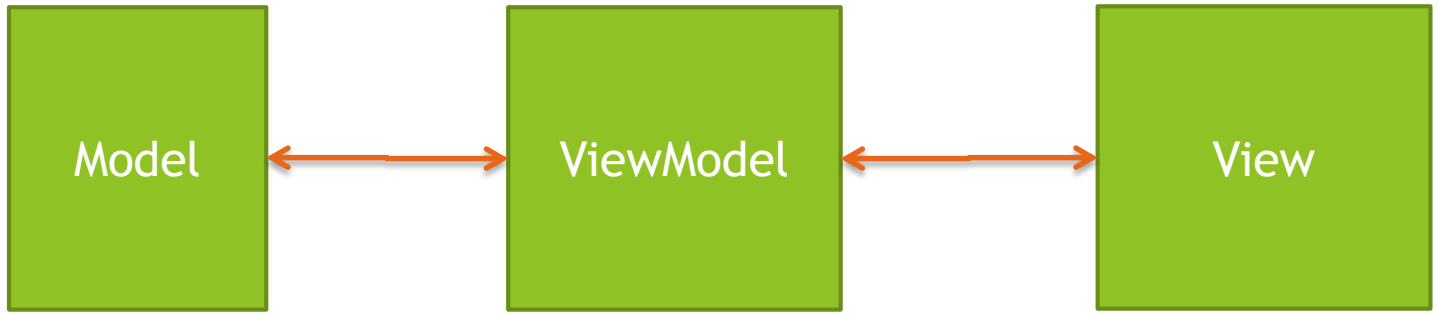
- Server Side
  - ASP.NET (4.6)
  - ASP.NET Core (1.1)
  - NodeJS
- Client Side ES5
  - KnockoutJS
  - Angular 1
- Client Side ES2015/2016
  - Angular 2
  - React
  - **Aurelia**

Client-side versus Server Side

- Server Side
  - Server produces HTML with data from database.
  - Browser renders HTML
  - When a user sorts or filters data then new HTML is requested from the server.
- Client Side
  - Server serves HTML with placeholders.
  - Server serves data in JSON/XML format with data from the database.
  - Browser renders HTML
  - Browser executes JavaScript that receives data from server
  - Browser executes JavaScript that replaces the placeholders with the data
  - When a user sorts of filters data then JavaScript performs sorting or filtering or browser requests new data.

Client Side - MVVM

- Model contains objects and operations in the business domain.
  - can be exchanged in JSON or XML format with the server
- ViewModel contains code representation of the data and operations on the UI
  - is bound to the HTML via client side binding
- View is a visible, interactive UI representing the state of the view model
  - contains the HTML with declarative bindings



Retrieve Data from Server

- XMLHttpRequest
  - IE10 of higher has full support
  - Native browser support
  - Support for Cross-Origin Resource Sharing
- $.ajax / $.getJSON
  - Supported in all browsers
  - Support for JSONP
  - Easier syntax
- Fetch API
  - Supported in lastest browsers
  - Polyfill available for older browsers
  - Used in Aurelia Http Client!

Single Page Application

- Legacy
  - Web site consists of many pages
  - Example:
    - http://website.com/default.aspx
    - http://website.com/login.aspx
    - http://website.com/article3.aspx
  - Browser must load multiple scripts / css for each page
- Modern
  - Web site consist of a single (full HTML) page (and many partial pages)
  - Example
    - http://website.com/
    - http://website.com/#login
    - http://website.com/#article3
  - Browser loads every script/style sheet only once.

Part 3

Putting it all together

Important Design Patterns

- Dependency Injection
  - Not `new View(new ViewModel(new HttpFetch(new HttpClient))`
  - But `Container.Create<View>()`
  - With interfaces loosely coupled
  - Perfect for unit testing

Important Design Patterns

- Modular Programming
  - Use a set of highly decoupled, distinct pieces of functionality stored in modules
  - Not big frameworks or applications but loosely coupled modules
  - Options
    - Async Module Definitions (AMD) (define/require)
    - CommonJS (exports/require)
    - SystemJS (register)
    - ES2015 modules (export)
  - Bundling and minification
  - HTML5 templates with combined JS and CSS!

Important Design Patterns

- Convention over Configuration
  - Don’t ask yourself why it works!
- Asynchronous programming
  - Every async function should return a Promise.

Module Package & Build Management

- Packaging
  - Ruby - Gem
  - .NET – NuGet
  - JavaScript - Bower
  - NodeJS – NPM
- Building
  - MSBuild
  - Grunt
  - Gulp
  - NPM scripts

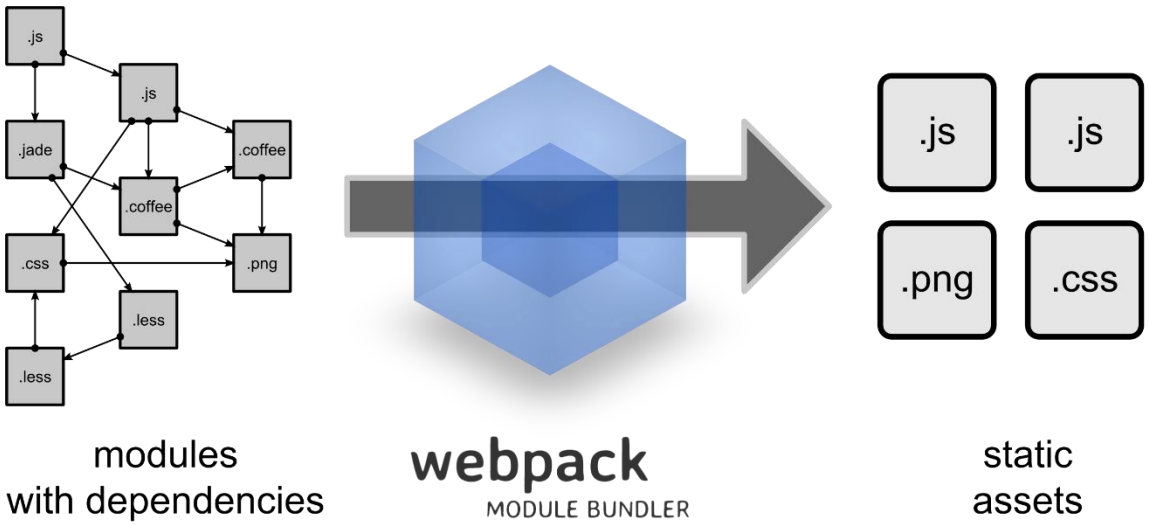
Bundling & Minification

- ASP.NET MVC with WebGrease
- Visual Studio 2015 Plugin: bundling & minification
- Via build system, e.g. Gulp
- WebPack

NodeJS

- Run JavaScript on server side
- Can be used as a web server
- Can be used for development tooling
- NPM package manager
  - package.json
  - TypeScript (2.0+) support for definitions!

WebPack


.js
.js
.jade
.coffee
.coffee
.css
.png
.less
.less
modules
with dependencies
webpack
MODULE BUNDLER
.js
.js
.png
.css
static
assets

WebPack

- Bundling, optimizing and minification
- Uses package.json from NPM
  - Dependency checking via package.json
- Modules
  - AMD, CommonJS, ES2015 (>2.0)
- Plugin model
  - HTML
  - CSS
  - SASS/LESS
  - And many others
- Versions
  - 1.4
  - 2.1.beta27
- WebPack Development Server

Part 4

Aurelia

Aurelia

- New SPA framework
- 1.0 version released 27-6-2016

- Consists of many libraries and plugins
  - you can only use want you want
  - Dependency Injection at the heart

- Retrieve via NPM
  - NPM registry hold the dependencies between Aurelia libraries

- Component based
  - Consist of View/ViewModel pairs

Aurelia - Build

- Build Systems
  - CLI
    - Own CLI: au
  - JSPM
    - Started with JSPM, many issues
    - Still compatible
  - WebPack 2
    - Added as alternative to JSPM
- Build on ES2015
  - Babel
  - TypeScript

Aurelia - Parts

- Dependency Injection
- Components
  - Convention over configuration
- Routing
  - Router and routing parameters
- Data binding
  - Templates
  - ${}
- Validation
  - Aurelia Validation

Part 5

RecipeWeb Demo

https://github.com/arjendeblok/ReceptenWeb

Part 6

Back to the Server Side

(ASP).NET Core

- Completely new development stack
- Multi-platform
  - Windows
  - Linux
  - Mac
- 1.1 version just released
- Use Visual Studio versions or command line dotnet.exe
- Uses same design patterns as Aurelia: DI, modules, convention over configuration.
- Independent of IIS. Runs on Kestrel web server. All requests run through the ASP.NET pipeline.
- Integrates MVC and Web API.

ASP.NET Core JavaScript Services

- Extends .NET with running JavaScript
- Calls NodeJS from ASP.NET pipeline
- Usages
  - Server-side data-binding for SEO
  - Development tooling like WebPack
  - Any other tool that runs under Node
- For Aurelia with WebPack
  - WebPack development integration
  - Web API and Web Site on same URL

Part 7

Testing

Testing

- Unit Testing
  - Test Runner
    - Karma Test Runner
  - Test Framework
    - Jasmine
    - Mocha
  - Stubbing
    - Sinon
      - 2: only in alpha

DEMO

Testing

- End to end testing
  - Protractor

Aurelia Menu

Voorgerechten

- HTML5
- CSS (SASS)
- Responsive Web Design
- JavaScript, ES 2015
- REST, Web API, CORS
- Dependency Injection
- Convention Based Programming
- TypeScript
- Promises
- Bundling & Minification
- NodeJS, NPM
- WebPack

Hoofdgerechten

- Aurelia met CLI en Babel
- Aurelia met JSPM en Babel
- Aurelia met WebPack, TypeScript en SASS

Nagerechten

- UI Framework
- Unit Testing
- E2E Testing
- Code Coverage
- Authentication
- UI Virtualization
- Validation
- Animation
- Mobile Apps (Cordova)
- Desktop Apps (Electron)

Links

- https://nodejs.org/en/
- https://www.typescriptlang.org/
- http://aurelia.io/
- https://webpack.github.io/
- https://twitter.com/AureliaEffect
- http://materializecss.com/
- https://github.com/arjendeblok
- https://github.com/arjendeblok/ReceptenWeb

THANK YOU