

ITC003A

Building web apps with Angular





Prerequisite

- HTML / CSS
- Javascript
- Typescript (Basic knoweldge)



Agenda

1 Introduction to Angular

2 Angular Core Concepts

3 Components and binding

4 Directives and elements

5 Dependency Injection

6 Pipes

7 Angular Routing

8 Angular Forms Module

9 Reactive programming with RxJS

10 Making Http Requests

1

Introduction to Angular

Evolution of web apps, single page apps, Angular framework and ecosystem



Evolution in Web App Development

Web sites

- ◉ Displaying information (text, images, ...)
- ◉ Hyperlinks to other pages
- ◉ No real interaction
- ◉ What HTML and HTTP were built for
- ◉ Heavy focus on the server

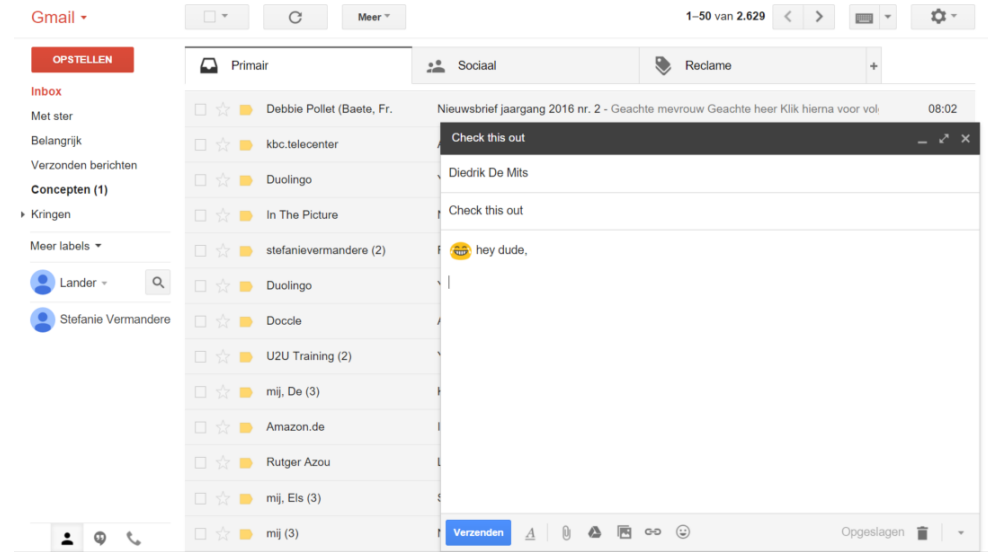




Evolution in Web App Development

Web apps

- A lot of interaction with the client
- Focus on the client (JavaScript!)
- Server is primarily for web services
- Modern standards for HTML/Javascript,... are evolving in this direction

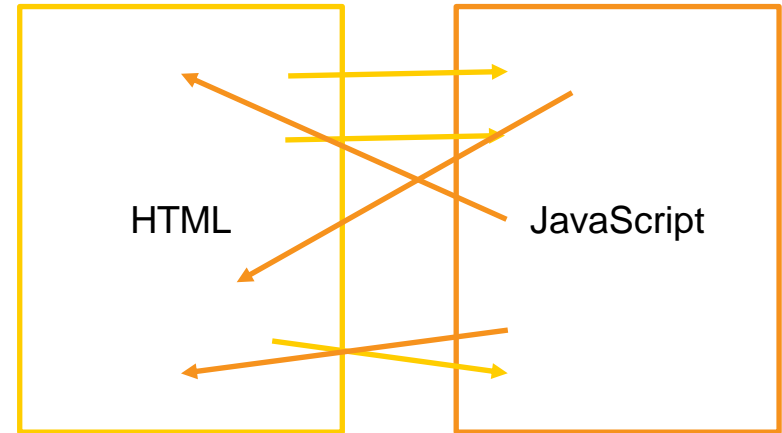




Web apps

First version

- Heavy intertwined code
- HTML and JavaScript impact each other
→ change in HTML leads to change in JS
- Bad for maintenance, testing, ...
- Native JavaScript and jQuery





Web apps (jQuery example)

```
function createOptions() {  
  $.each(availableDates, function (key, date) {  
    var value = date.toLocaleDateString();  
    $('#changeDate')  
      .append($("<option></option>"))  
      .attr("value", value)  
      .text(value);  
  });  
};  
  
$('#update').click(function () {  
  availableDates = getDates();  
  createOptions();  
});
```

DOM selection

HTML
generation in JS

Event Handling



Web apps

Second version

- Separated Code thanks to Data Binding → both can evolve without heavy impact on the other
- Better for maintenance, testing, ...
- Backbone, Knockout, AngularJS 1, Ember





Web apps (AngularJS example)

```
<input type="button" ng-click="addBooking()" value="Book!" />
<table>
  <thead><tr><th>Artist</th><th>Date</th></tr></thead>
  <tbody>
    <tr ng-repeat="booking in bookings">
      <td>{{booking.artist}}</td>
      <td>{{booking.date | date:shortDate}}</td>
    </tr>
  </tbody>
</table>
```

HTML

```
app.controller('BookingsController', function ($scope) {
  $scope.current = new Booking('Jamie Cullum', $scope.availableDates[1]);

  $scope.bookings = [];

  $scope.addBooking = function () {
    $scope.bookings.push($scope.current);
    $scope.current = new Booking('', new Date());
  }
})
```

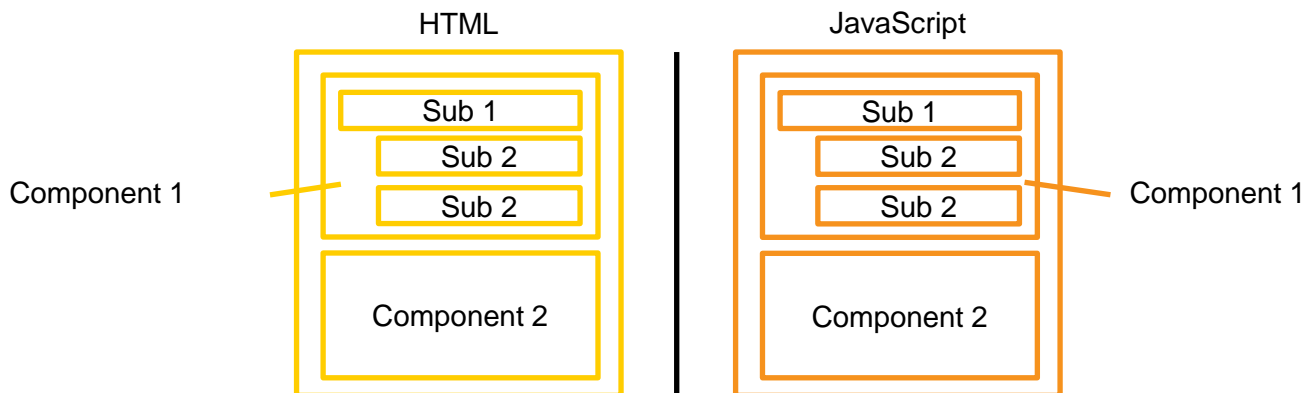
Javascript



Web apps

Third version

- Only using Data Binding can lead to very large and complicated code
- Componentizing Apps deals better with complexity
- Focus on latest standards for HTML and JS





Single Page App (SPA)

What is a SPA ?

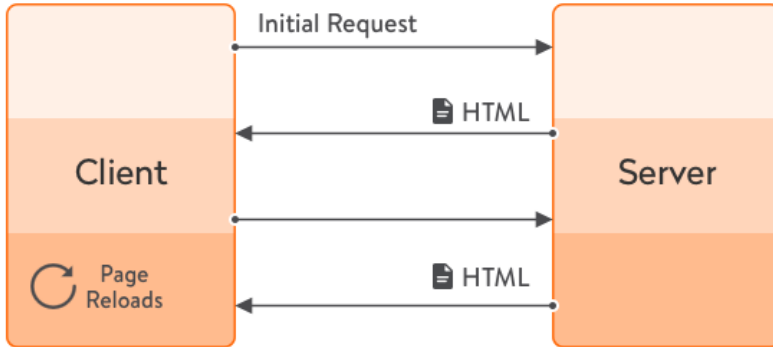
- A next evolution from traditional website
- Web apps that load a single HTML page
- Client side (JavaScript) based rendering
- Support for multiple platforms



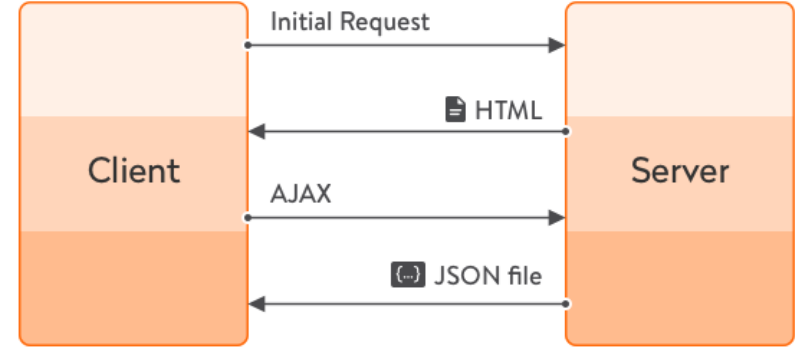


Single Page App (SPA)

Multi-page app lifecycle



Single-page app lifecycle





Single Page App (SPA)

Benefits

- App-like user experience
- Faster UI, more interactivity
- Be able to work offline
- Support for multiple platforms
- Offloading the server responsibility



Angular Framework

What is Angular ?

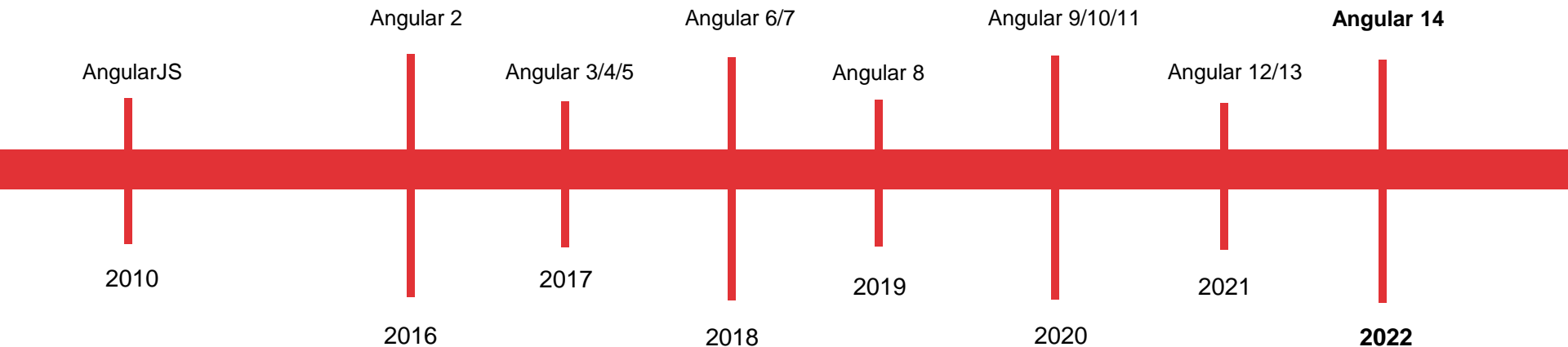
- Angular is a JavaScript framework
- Developed and supported by Google
- Build single page applications
- Angular code is written in Typescript
- Based on data binding
- Built-in dependency injection mechanism





Angular Framework

History



Angular 2 is **NOT** a new version of AngularJS
It's a complete rewrite



Angular Language Support

- Angular want new features, but also want to support older browsers as well
- Angular support three possibilities
 - Typescript
 - Dart
 - Javascript
- Currently **TypeScript** is the default language for Angular development
- Uses **Webpack** to transpile Typescript into Javascript (+ other tasks)



Angular Stylesheet Support

- Angular supports pure **CSS**, and CSS **preprocessor*** like **SASS**, **LESS** or **PostCSS**
- SASS is the default preprocessor used by Angular
- SASS stands for **S**yntactically **A**wesome **S**tyle **S**heets
- SASS is CSS **preprocessor*** that provides unique features such as variables, nested rules, inline imports, and more.
- Sass is compatible with all versions of CSS

* A **CSS preprocessor** is a program that lets you generate CSS from the preprocessor's own unique syntax.



Angular Framework

Building blocks

- An application is a group of modules
- A **module** contains components
- Basic building block of Angular is **component**
- A component can use **services**
- A module can use or embed other modules



Angular Framework (Architecture)

Application

Module

Component

Component

Component

Service

Service

Module 1

Module 2

Module 3