


The background features the AngularJS logo, which is a large, stylized letter 'A' composed of two overlapping triangles. The 'A' is centered and rendered in a dark gray color. It is set against a black background that has a faint, larger-scale version of the same 'A' shape, creating a layered effect.

Vývojařská Plzeň

AngularJS



Milan Lempera

@milanlempera

PHP -> Javascript



Vít'a Plšek

@winsik

PHP -> Java -> Javascript



www.angular.cz



?

Psal se rok 2009 ... ano ... 2009 ...

Jak se tehdy tvořili webové aplikace ?

- pomocí server-side frameworků
- scriptování v jQuery

Původním cílem AngularJS byl komerční framework

- náhrada stávajících view
- příklon k **Single Page aplikacím**

“I often say that AngularJS is what the web browser would have been, had it been designed for application”

Miško Hevery

AngularJS Creator

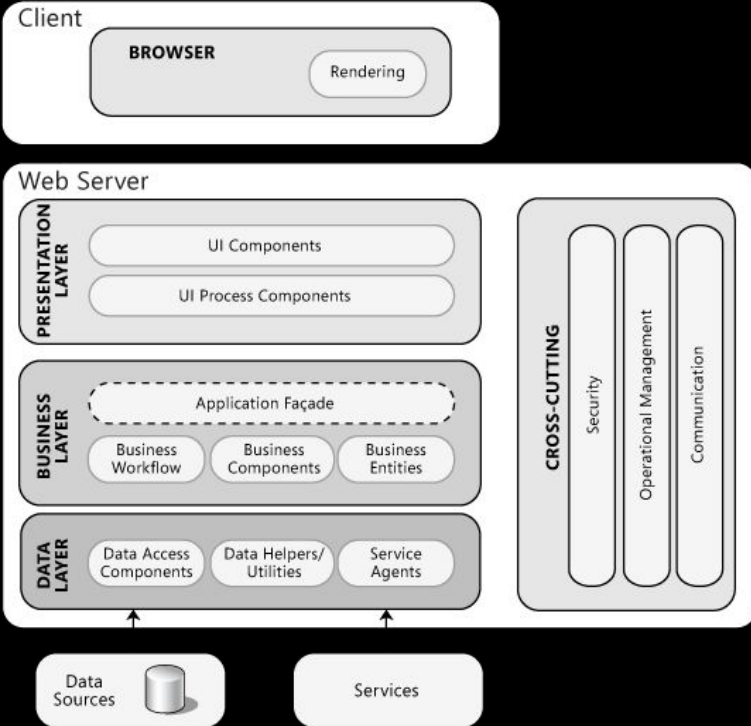


Serverside

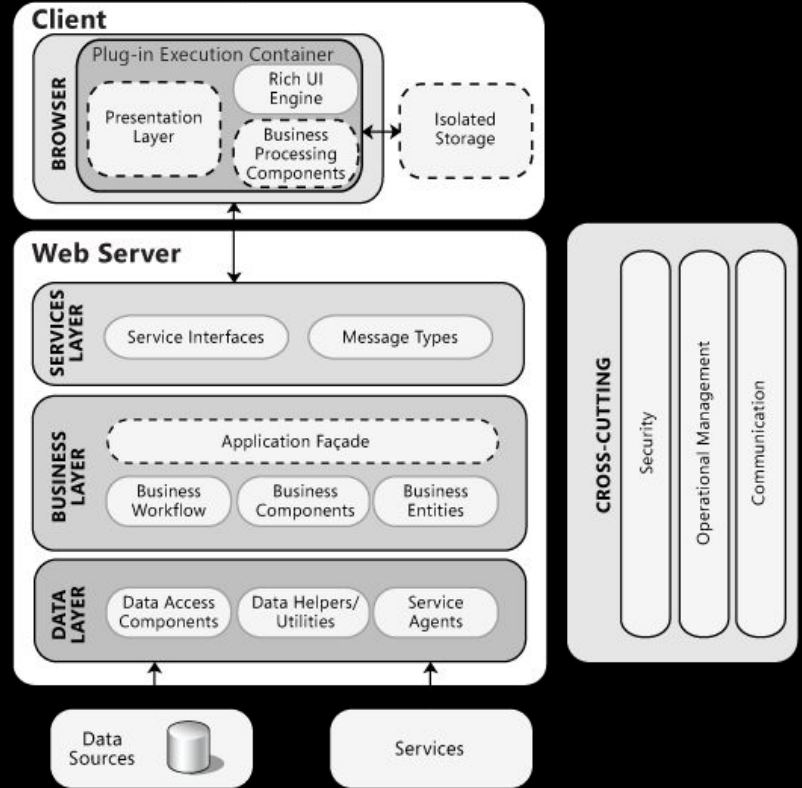
VS

SPA

Web Application



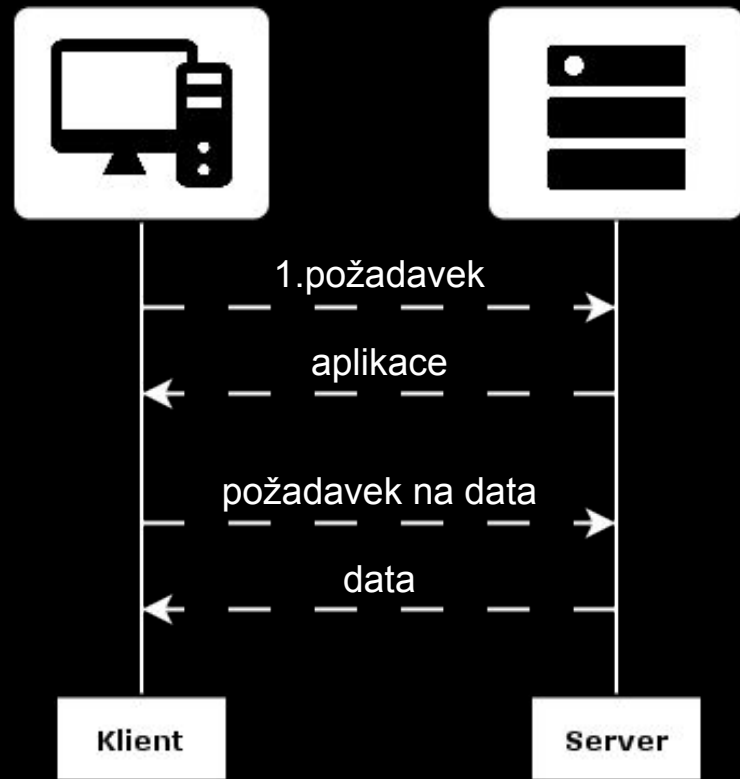
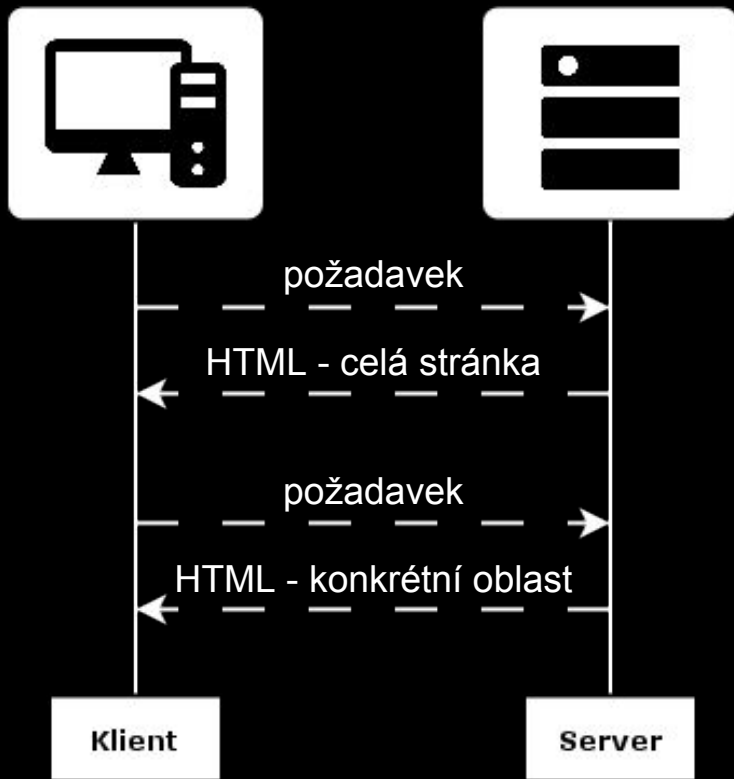
Rich Internet Application



Serverside

VS

SPA



Šablonový systém v AngularJS

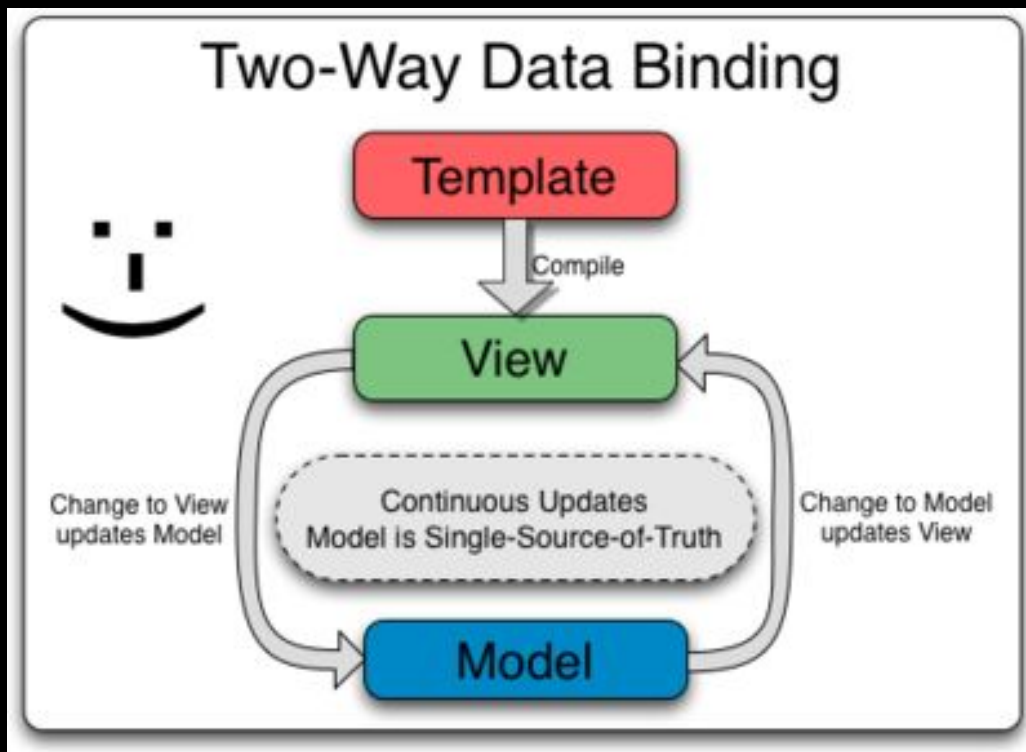
```
<html ng-app>  
  <head>  
    <script src="https://...../angularjs/1.4.8/angular.min.js"></script>  
  </head>  
  <body>  
  
    <h1>Ahoj {{city.name}}</h1>  
    Upravit oslovení: <input type="text" ng-model="city.name" />  
  
  </body>  
</html>
```


Directive

```
<span ng-if="user.enabled">  
  {{user.name}}  
</span>
```

```
<tr ng-repeat="user in users">  
  <td>{{user.name}}</td>  
</tr>
```

Dvoucestný databinding - milovaný i nenáviděný



Dvoucestný databinding - milovaný i nenáviděný

```
<h1>Ahoj {{city.name}}</h1>
```

```
Upravit oslovení: <input type="text" ng-model="city.name" />
```

Watcher
{{city.name}}
city.name (ngModelWatch)

Dvoucestný databinding - milovaný i nenáviděný

`<h1>Ahoj {{city.name}}</h1>`

Upravit oslovení: `<input type="text" ng-model="city.name" />`

Watcher	last value	curent value	changed?
<code>{{city.name}}</code>	""	"Plzeň"	true
<code>city.name (ngModelWatch)</code>	""	"Plzeň"	true

Dvoucestný databinding - milovaný i nenáviděný

`<h1>Ahoj {{city.name}}</h1>`

Upravit oslovení: `<input type="text" ng-model="city.name" />`

Watcher	last value	curent value	changed?
<code>{{city.name}}</code>	“Plzeň”	“Plzeň”	false
<code>city.name</code> (<code>ngModelWatch</code>)	“Plzeň”	“Plzeň”	false

Dvoucestný databinding - jak s ním žít

- angular výrazy vyhodnocuje pokud se něco mohlo změnit
 - o musí být rychlé
 - o nemáte pod kontrolou počet volání
- počet watcherů < ~2500

- + výměnou za to můžete pracovat přímo s JS objekty
- + ve většině případů vám ušetří čas a kód

Formuláře

```
<form name="breweryForm">
```

```
  <input id="brewery-year"  
    type="number"  
    name="year"  
    ng-model="brewery.year"  
    required />
```

```
</form>
```

```
breweryForm.year = {  
  $dirty: false,  $pristine: true,  
  $valid: false,  $invalid: true,  
  $touched: false,  
  $untouched: true,  
  $error: {  
    "number": true  
    "required": true  
  }  
}
```

Dependency injection

```
var UserStore = function() {  
  this.httpService = HTTPService.getInstance();  
}
```

```
var UserStore = function(httpService) {  
  this.httpService = httpService;  
}
```


Dependency injection v angularu

```
var BreweryServiceConstructor = function() {  
    // služba řeší komunikaci se serverem  
}
```

```
var BreweryController = function(breweryService) {  
    this.breweries = breweryService.load();  
};
```

```
angular.module("beerApp", [])  
    .controller("BreweryController", BreweryController)  
    .service("breweryService", BreweryServiceConstructor);
```

Komunikace se serverem

\$http - základní rozhraní

\$resource - jednoduché pojetí RESTu

```
var Brewery = $resource('/api/v1/brewery/:id', {id : '@id'});
```

```
var list = Brewery.query({'page' : 2}); // GET /api/v1/brewery
```

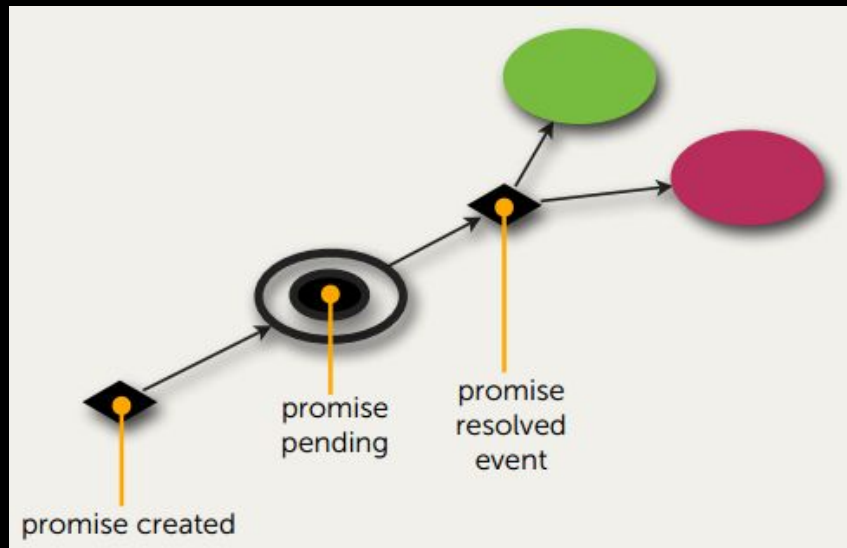
```
var brewery42 = Brewery.get({id : 42}); // GET /api/v1/brewery/42
```

```
brewery42.name = 'Janáček';
```

```
brewery42.$save(); // POST /api/v1/brewery/42
```

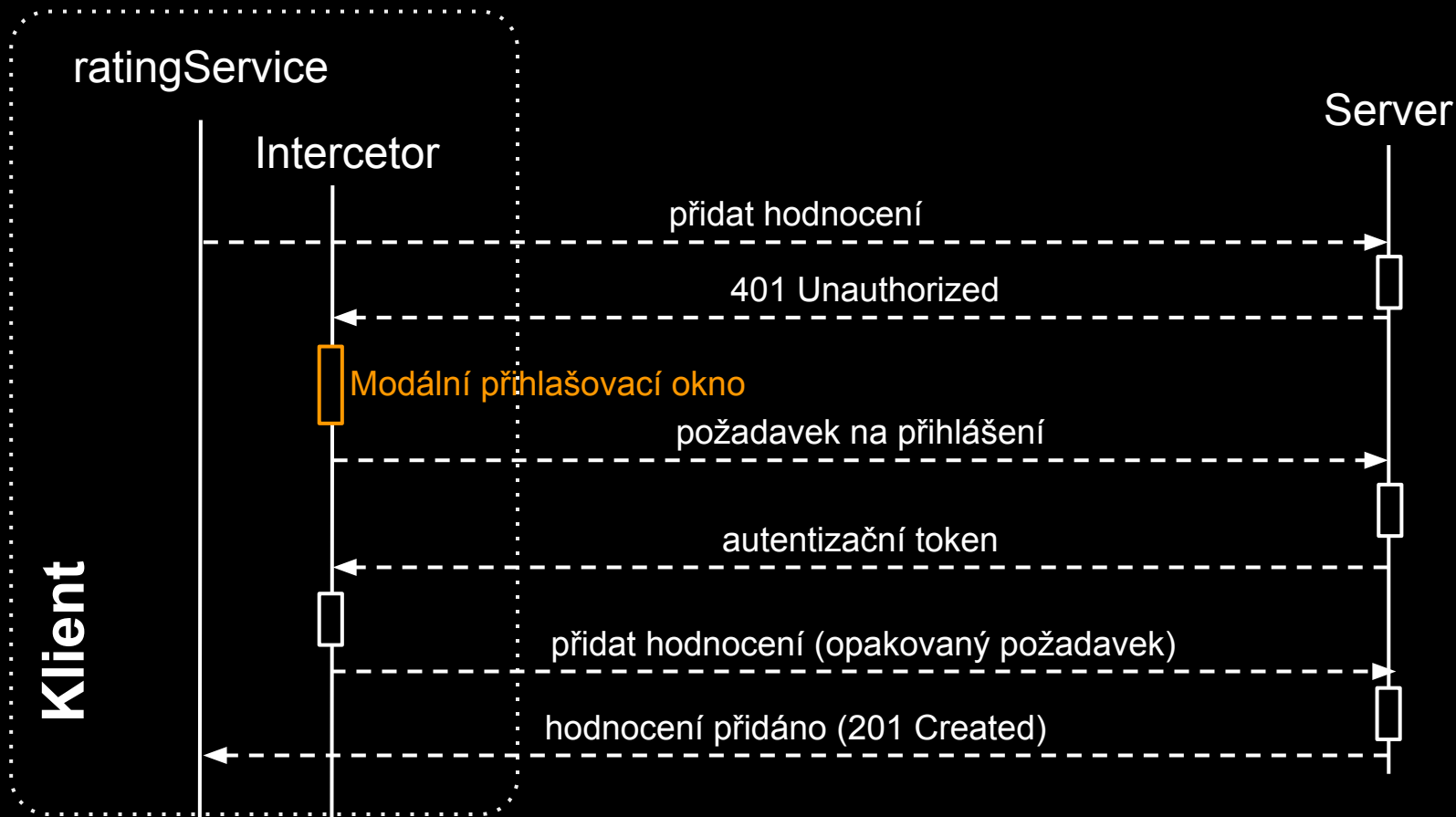
```
brewery42.$delete(); // DELETE /api/v1/brewery/42
```

Promise



```
functionWithPromise()  
  .then(promisedStep2)  
  .then(function (value2) {  
    // Do something  
  })  
  .catch(function (error) {  
    // Handle error from  
    // all above steps  
  })
```

Interceptory



Routing

Beer database

beer-app.angular.cz/#/brewery/edit/48

`#/brewery`
controller: BreweryListController,
templateUrl: brewery/list.html

`#/brewery/edit/:id`
controller: BreweryEditController,
templateUrl: brewery/edit.html

index.html

Header

ng-view

brewery/list.html

nebo

brewery/edit.html

Vlastní elementy - directives

potřebujeme

- předat data
- reagovat na akce

```
<brewery-form  
  brewery="....."  
  on-save="....."  
  on-cancel=".....">  
</brewery-form>
```

Vlastní elementy - directive

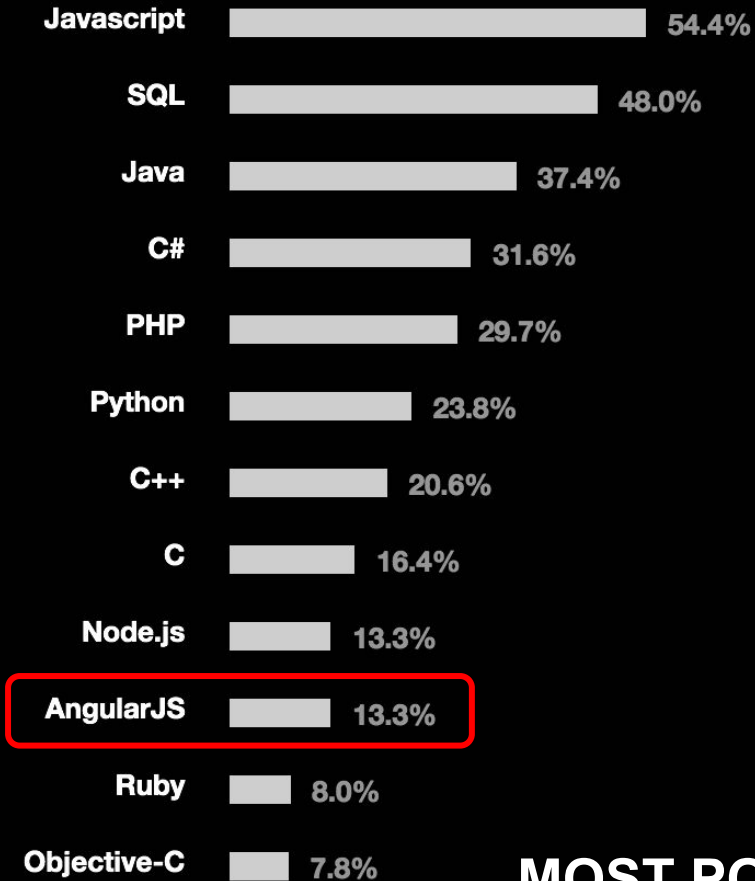
```
var directiveDefinitionObject = {  
  templateUrl: "breweryForm.html",  
  controller: controllerConstructor  
  bindToController: {  
    brewery: '=',  
    onSave: '&',  
    onCancel: '&'  
  }, ...  
};
```

```
<brewery-form  
  brewery="....."  
  on-save="....."  
  on-cancel=".....">  
</brewery-form>
```

```
angular.module('beerApp.breweryForm', [])  
  .directive("breweryForm", function() {  
    return directiveDefinitionObject;  
  })
```

Proč si angular vybrat

- Výborná dokumentace
- Početná komunita
- Ucelený framework
- Oddělení kódu a html
- Testovatelnost
 - ngMock
 - protractor
- Velké množství komponent a knihoven
- **AngularJS přerostl své odvětví**



MOST POPULAR TECHNOLOGIES

Stack Overflow - Developer Survey 2015

Kdy ho použít?

Aplikace, aplikace aplikace

- informační systémy
- kalkulační nástroje

- ... téměř cokoli za přihlášením

Tam kde potřebujete vyměnit view a dokážete postavit REST API

Kdy ho nepoužít?

Z důvodu SEO

- klasické webové stránky
(není náhrada jQuery)
- inzerce, e-shopy ?

Z důvodu výkonu

- vysoce dynamické aplikace
- “Excel”

Na co si dát pozor?

- množství a rychlost watcherů
 - více aplikací na jedné stránce
-
- pište čistý kód
 - malé části kódu řešící jedem problém
 - nespolehejte na odstínění od javascriptu
 - prototypová dědičnost
 - promise
 - this

Kdo ho používá

The Google logo is displayed in its standard multi-colored font: blue 'G', red 'o', yellow 'o', green 'g', and red 'le'.The SEZNAM.CZ logo is shown in a white, stylized, handwritten font. The 'S' is significantly larger and more decorative than the other characters.

Youtube, PayPal, Mall, GoG, InvisionApp, Blue Origin, Lego
Indiegogo, Docker, Best buy, NBA, Forbes, CNN,
Weather...

<https://www.madewithangular.com>

Budoucnost - 1.x vs 2.0

- důvod ?
- technologie ?
- datum vydání ?
- upgrade ?

Má smysl na něj čekat?

- 2.0 bude úplně nový framework ...





Díky za pozornost ...

... a **nešetřete** záludnými dotazy.

Zdroje - www.angular.cz/plzen-2015

