

# AngularJS

framework pro moderní  
webové aplikace

24.března 2015 - CZJUG Brno

**Milan Lempera** @milanlempera

**Vít'a Plšek** @winsik



[www.angular.cz](http://www.angular.cz)

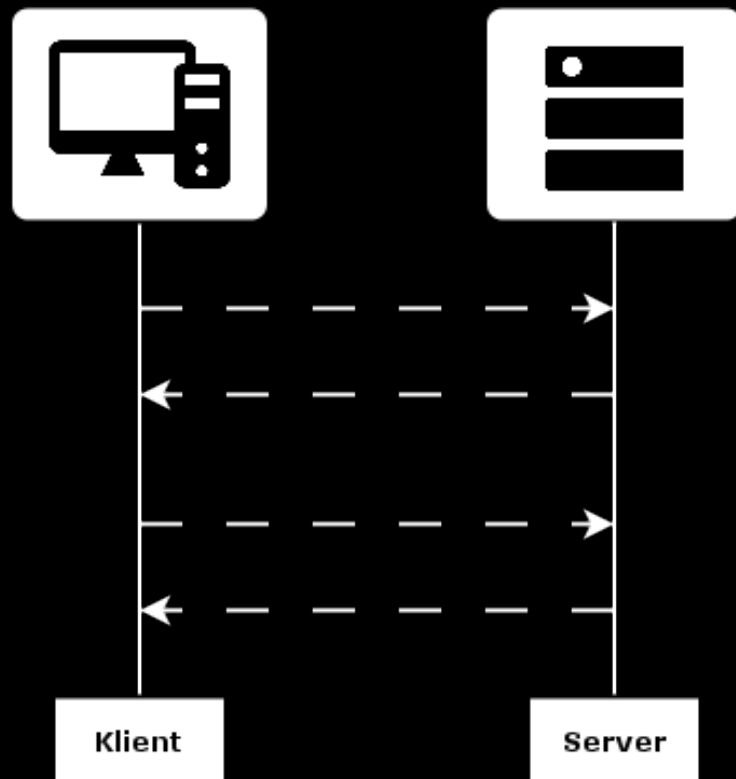
**Když se řekne webová aplikace ...**

Bez jaké technologie to určitě nepůjde ?

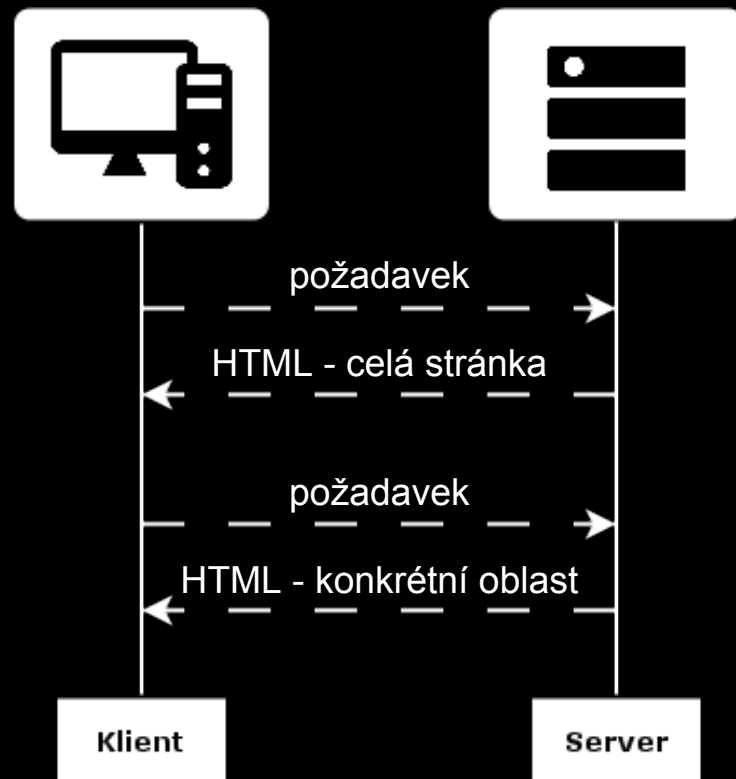
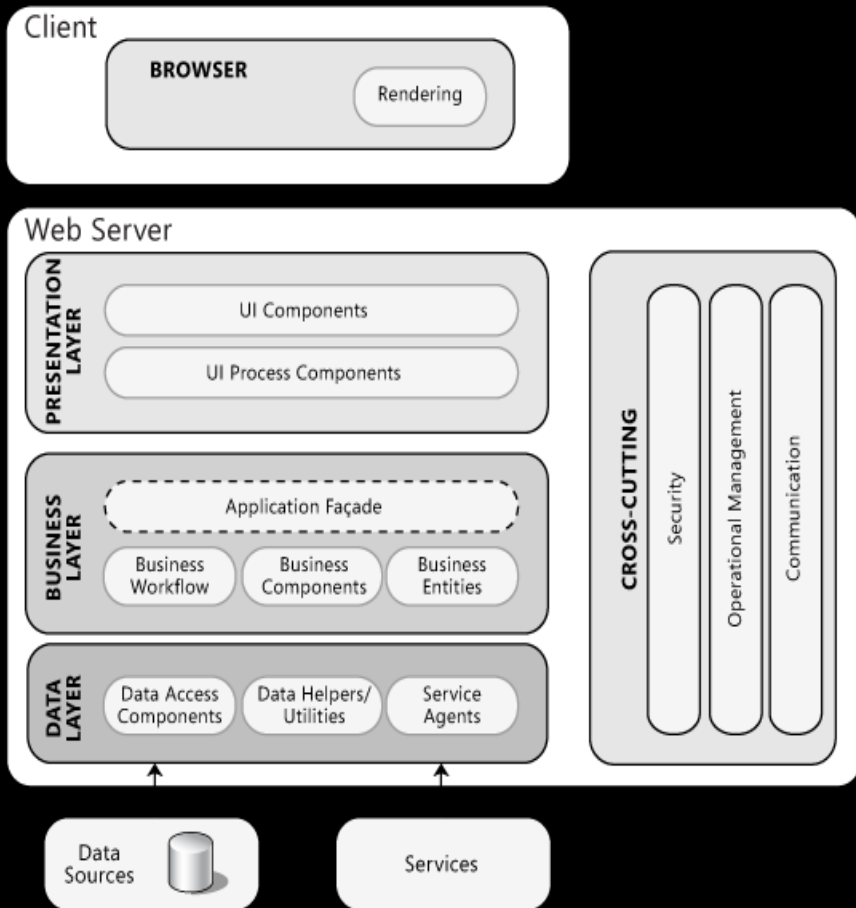
**HTML**

# To se pak zobrazuje v prohlížeči ...

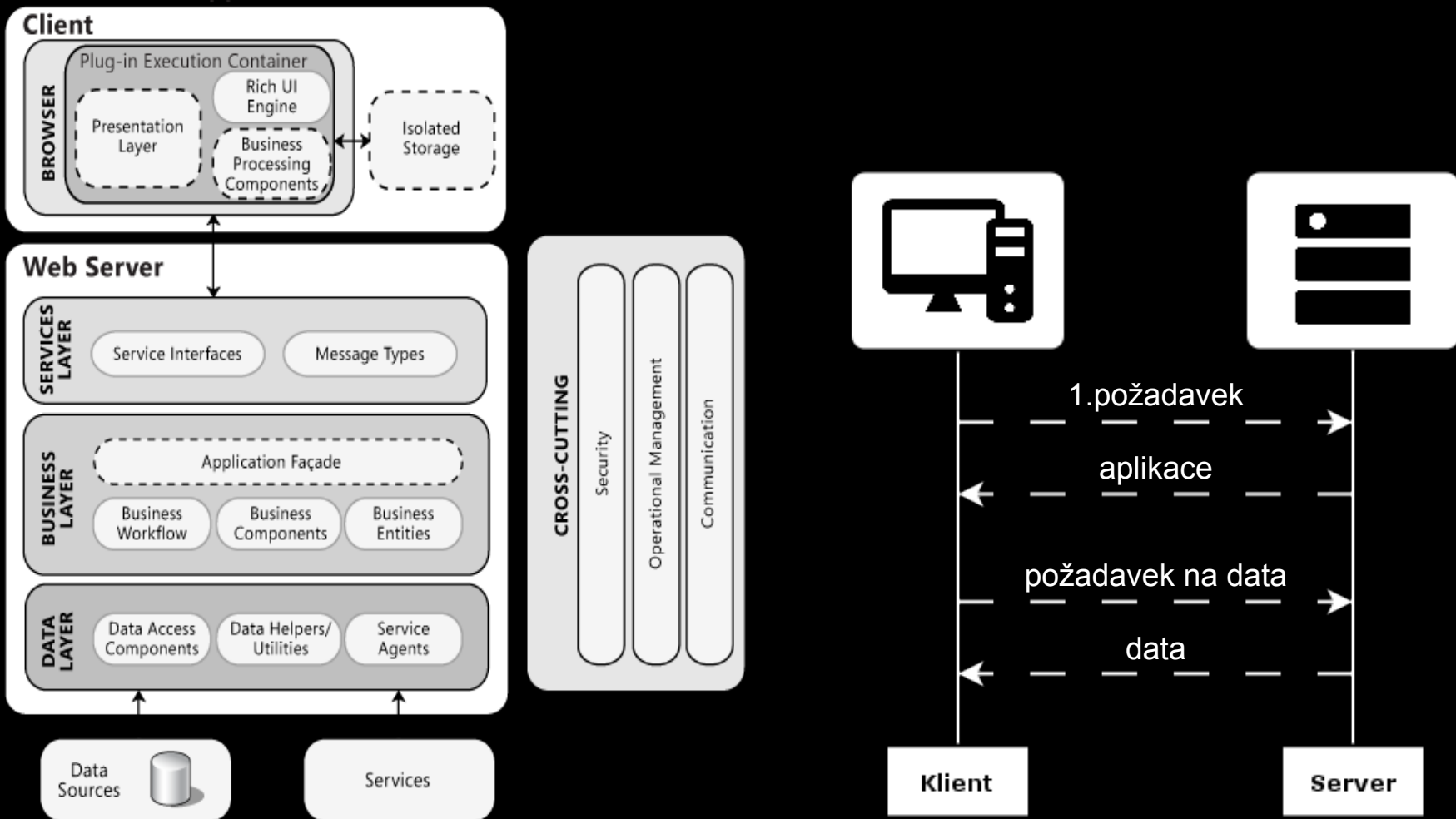
Ale jak se tam dostane?



# Web Application



# Rich Internet Application



**Co umožňuje tvořit takové aplikace?**

**JAVASCRIPT**

# Javascript

**1995 - Netscape**

GTS CzechCom 14,4 kbit/s 10 500 Kč

**1996 - Microsoft - JScript**

**1997 - ECMAScript**

# Javascript

## ECMAScript 5/5.1 (2009/2011)

- IE9\*, FF, Chrome

## ECMAScript 6 / ECMAScript 2015





# Web není desktop ...

**AUTO KELLY**  
vše pro auta

Přihlásit se | Registrace

PRO MOTORISTY PRO SERVISY VYBAVENÍ SERVISŮ

Nákupní košík  
0 ks zboží, 0 Kč s DPH

Katalog

Kód Auto Kelly:  Rozšířené vyhledávání

**Náhradní díly - osobní auta**

**Univerzální díly**

Stěrače

Ramínkové

Sady

700mm+650mm

**Náplně**

Autobaterie

Autopříslušenství

Autochemie a autos kosmetika

Kola, disky, příslušenství

Pneumatiky

Skútry a elektrokola


Vše pro dílnu

Reklamní předměty a oblečení

Náhradní díly - nákladní auta

Náhradní díly - nápravy

**ZÁKAZNICKÁ LINKA**


 **840 808 808**

71 PRODEJEN

**CHAMPION AEROVANTAGE KIT sada 700mm + 650mm**

**VÝPRODEJ**

Objednávací kód: CH K70A/C02



www.autokelly.cz

Vaše cena bez DPH: **1 196 Kč**  
Vaše cena s DPH: **1 448 Kč**  
Zaruka (v měsících): **24**

Dostupnost v CS: **Skladem**  
Dostupnost v jiné pobočce: **Also**  
Dostupnost v e-termín skladu: **Ne**

Zobrazit všechny pobočky

Množství:

Popis produktu Kódy Náhrady

**INFORMACE O PRODUKTU**


balení: sada  
byp. ramínkový klasický stěrač - konstruktivně shodný s OE  
délka: 1x délka 2 (jmén): 700 / 650

**INFORMACE O VÝROBCI**

CHAMPION

Značka CHAMPION je jednou ze značek nabízených koncernem FEDERAL MOGUL. V portfoliu Auto Kelly naleznete pod značkou Champion - filtry, stěrače a zapalovací svíčky.  
FEDERAL MOGUL patří mezi přední dodavatele nejen pro aftermarket, ale také do výroby pro VV GROUP, Peugeot, Citroen, Renault, Ford, GM a mnoho dalších.

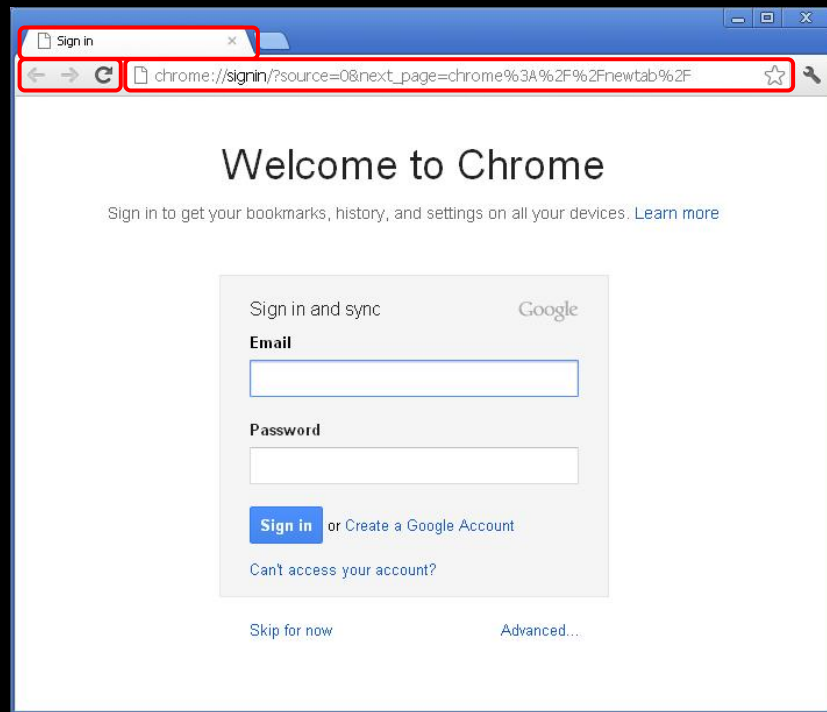
Pokud chcete více informací o dodavateli www.federal-mogul.com/en.



www.autokelly.cz

# Web není desktop ...

- běhovým prostředím je webový prohlížeč
  - tlačítka zpět, vpřed
  - adresní řádek
  - taby



# Web není desktop ...

... rozhodli jsme se přesunout tlačítko Zpět zprava na levou stranu tak, jak jste zvyklí ze svých prohlížečů ....

...když i přesto náhodou kliknete na Zpět v prohlížeči, neodhlásíme vás z internetového bankovníctví hned, jako jsme to dělali dřív...

The screenshot shows the Air Bank web interface. At the top, there's a navigation bar with buttons for 'ÚČTY', 'ZAPLATIT', 'PŮJČKY', 'PŘEHLEDY', and 'NASTAVENÍ'. Below this, account balances are displayed for BU1 CZK (4 750,92 CZK), SU1 CZK (4 750,92 CZK), and SU1 USD (1039023036). A table shows recent transactions with columns for date, description, and amount. On the right, there's a section for 'Nezpracované platby' (Unprocessed payments) with items like 'Neúspěšné platby' and 'Platby ve zpracování'. A red box highlights the back button in the browser, and a red callout bubble says 'Chcete zpět?' (Do you want back?).

BU1 CZK	4 750,92 CZK	SU1 CZK	SU1 USD
1039023044	4 750,92 CZK	1039023028	1039023036

Poslední změny na vašem účtu	Peníze úročíme 300xovou rychlostí	Nezpracované platby
14.11.2013 Air Bank Spánka půjčky: Půjčka 1		Neúspěšné platby 0,00 CZK
01.11.2013 Air Bank		Platby ve zpracování 0,00 CZK
01.11.2013 Air Bank		V příštích 7 dnech 0,00 CZK
21.10.2013 SU1 CZK 1039023028:0030		Nevyřazená inkasa 0,00 CZK
17.10.2013 SU1 CZK 1039023028:0030		



ANGULARJS  
by Google

HTML enhanced for web apps

since 2009

```
<html ng-app>
  <head>
    <script src="https://...../angularjs/1.3.14/angular.min.js"></script>
  </head>
  <body>

    <h1>Ahoj {{accost.name}}</h1>
    Upravit oslovení: <input type="checkbox" ng-model="accost.modify">

    <div ng-if="accost.modify">
      <label>Zadejte jméno:</label>
      <input type="text" ng-model="accost.name">
    </div>
  </body>
</html>
```

[ukázka](#)

# ng-co? ... vestavěné directive

ng-app

ng-model="accost.name"

ng-model="accost.modify"

```
accost: {  
  modify: true,  
  name: "Ostravo!!!"  
}
```

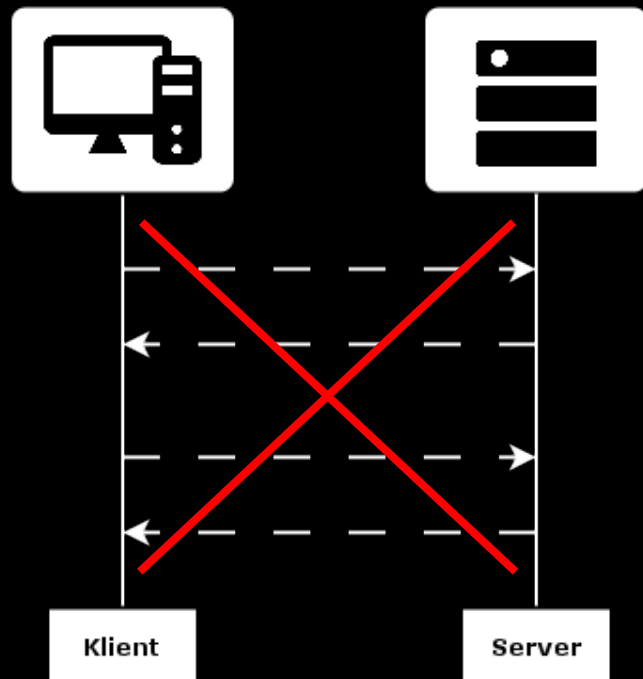
{{accost.name}}

ng-if="accost.modify"

# Celá aplikace opravdu běží u klienta?

Serverový framework

Klientský framework



# Zadání:

zobrazte seznam položek



```
<html ng-app="statusApp">
  <head>... </head>
  <body ng-controller="ReportController as reportController">
    <button ng-click="reportController.reportError()">Reportovat ...</button>

    <table class="table table-condensed">
      <thead>...</thead>
      <tbody>
        <tr ng-repeat="report in reportController.reports"
          ng-class="{ 'danger': report.status === false }">
          <td>{{report.date | date: 'medium'}}</td>
          <td>{{report.status ? 'OK' : 'ERROR'}}</td>
        </tr>
      </tbody>
    </table>

  </body>
</html>
```

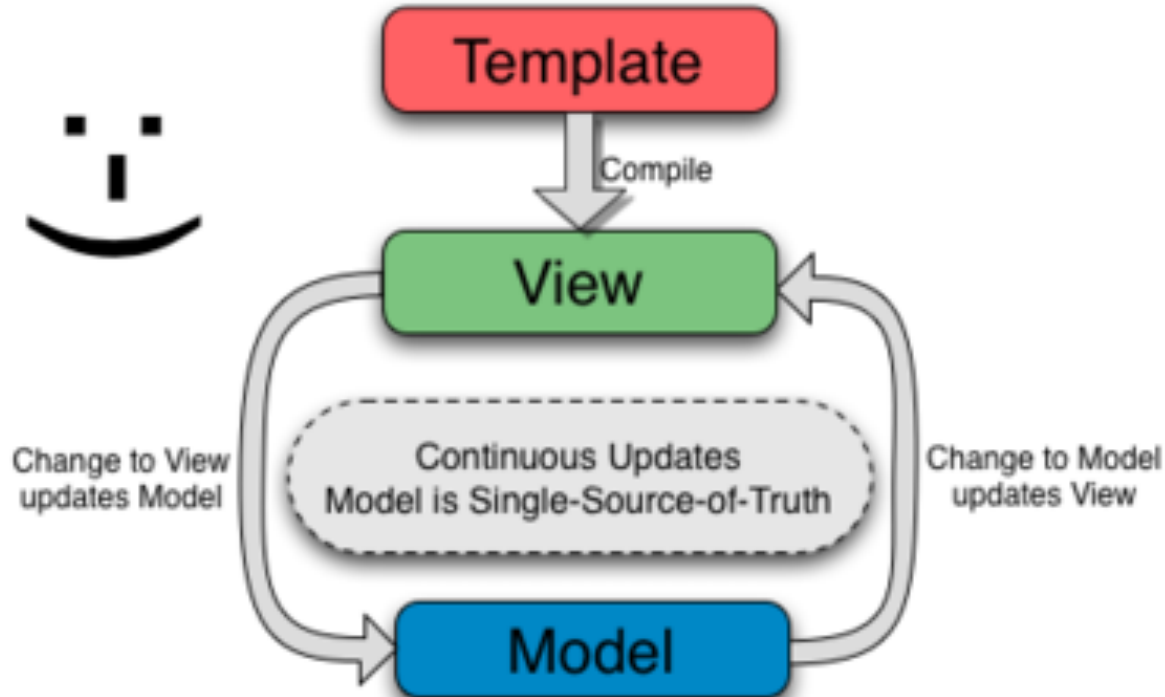
[ukázka](#)

```
var ReportController = function() {  
  this.reports = [  
    { date: new Date("2015-03-09 11:28:03"), status:true },  
    { date: new Date("2015-03-09 11:12:54"), status:false },  
    { date: new Date("2015-03-09 10:43:16"), status:true }  
  ];  
};
```

```
this.reportError = function() {  
  var newReport = { date: new Date(), status:false };  
  this.reports.push(newReport);  
};  
};
```

```
angular.module("statusApp",[])  
  .controller("ReportController", ReportController);
```

# Two-Way Data Binding



# Zadání:

editační formulář

```
<form name="breweryForm">
  <label for="brewery-name">Název</label>
  <input id="brewery-name" type="text"
    name="name"
    ng-model="brewery.name"
    required
  />

  <label for="brewery-year">Rok založení</label>
  <input id="brewery-year" type="number"
    ng-model="brewery.year"
    name="year"
    min="0"
  />

  <button type="submit">Uložit</button>
  <button type="reset">Zrušit</button>
</form>
```

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

```
// angular vytvoří objekt
```

```
breweryForm = {  
  $submitted: false,  
  $dirty:     false,  $pristine: true,  
  $valid:     false,  $invalid: true,  
  $error:    {required: Array[2]}  
}
```

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

```
<input type="number"
```

```
  ng-model="brewery.year"
```

```
  name="year"
```

```
  required
```

```
  min="0"
```

```
/>
```

```
breweryForm.year = {
```

```
  $dirty:    false,    $pristine:    true,
```

```
  $valid:    false,    $invalid:     true,
```

```
  $touched: false,    $untouched:  true,
```

```
  $error:    {"number": true}
```

```
}
```

[ukázka](#)

# Zadání:

chceme použít formulář na víc místech



# Jak to budeme používat?

potřebujeme

- předat data
- reagovat na
  - odeslání
  - zrušení

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

# Jak to vytvořit?

```
angular.module('beerApp.breweryForm', [])  
  .directive("breweryForm", function() {  
    return {  
      templateUrl: "breweryForm.html",  
      scope: {  
        brewery: '=',  
        onSave: '&',  
        onCancel: '&'  
      },  
      .....  
    };  
  })
```

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

[kompletní kód](#)

# Existující komponenty?

<https://angular-ui.github.io/>

<http://ngmodules.org/>

# Controller v angularu

inicializuje stavu  
přidává chování

# Dependency injection

# Dependency injection

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

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

# Dependency injection

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

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

# Dependency injection v angularu

```
var BreweryController = function(breweryService) {  
  this.save = function() {  
    breweryService.save(this.brewery)  
      .then(function(brewery) {  
        // do something with brewery  
      },  
      function(error) {  
        // show error  
      });  
  };  
};  
  
angular.module("beerApp", [])  
  .controller("BreweryController", BreweryController);
```



# Dependency injection v angularu

```
var BreweryController = function(breweryService) {.....};  
BreweryController.$inject["breweryService"];
```

```
var BreweryServiceConstructor = function() {...}
```

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

# Shrnutí

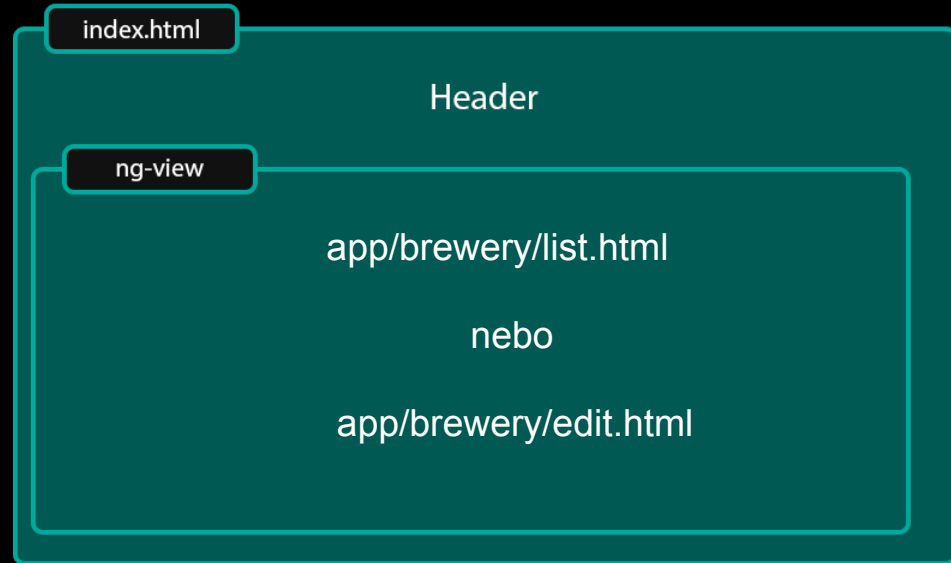
- rozšiřuje HTML
- model (data) se promítají do šablony
- dvoucestný databinding
  
- controller - stav a chování
- vlastní elementy (directivy)
- podpora DI (dělení logiky na malé části)

# Routing

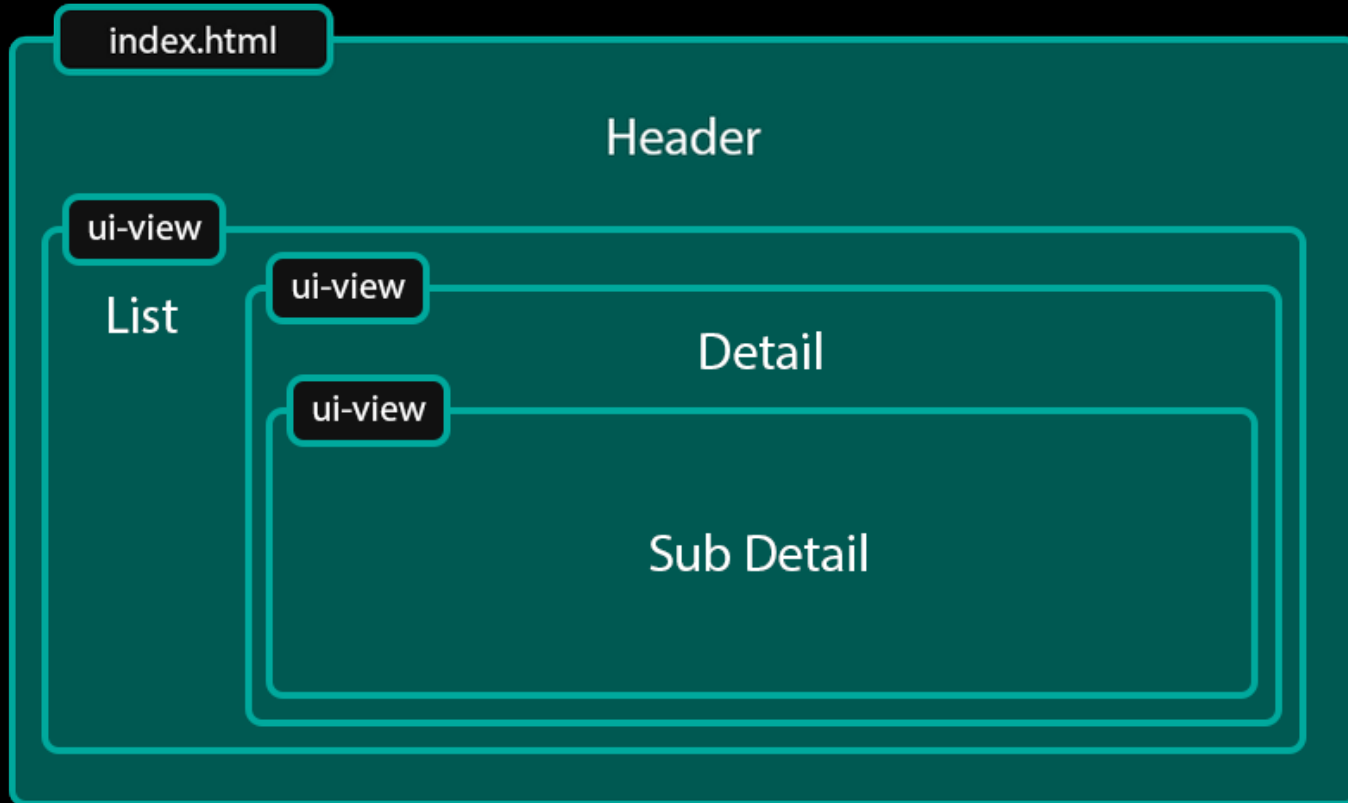
- transparentní chování aplikace v prostředí webu
- stav a URL adresa jsou vzájemně provázané
  - historie (vpřed, zpět)
  - možnost poslat/uložit URL

# Routing (ngRouter)

```
breweryModule.config( function($routeProvider) {  
  $routeProvider.  
    when("/brewery", {  
      templateUrl: "app/brewery/list.html",  
      controller: "BreweryListController"  
    }).  
    when("/brewery/create", {  
      templateUrl: "app/brewery/edit.html",  
      controller: "BreweryEditController"  
    }).  
    otherwise({  
      redirectTo: "/brewery"  
    });  
});
```



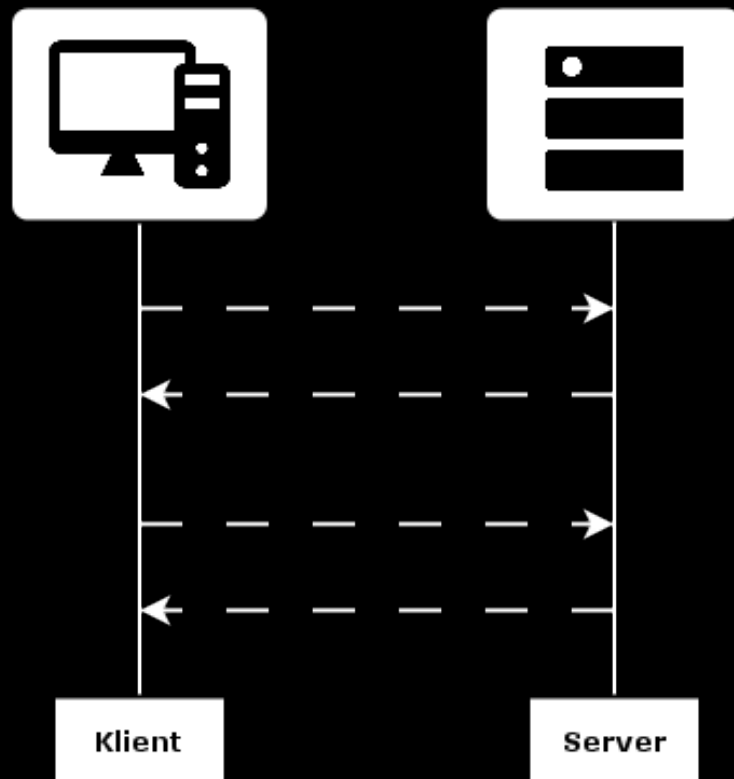
# Routing (uiRouter)



# Komunikace se serverem

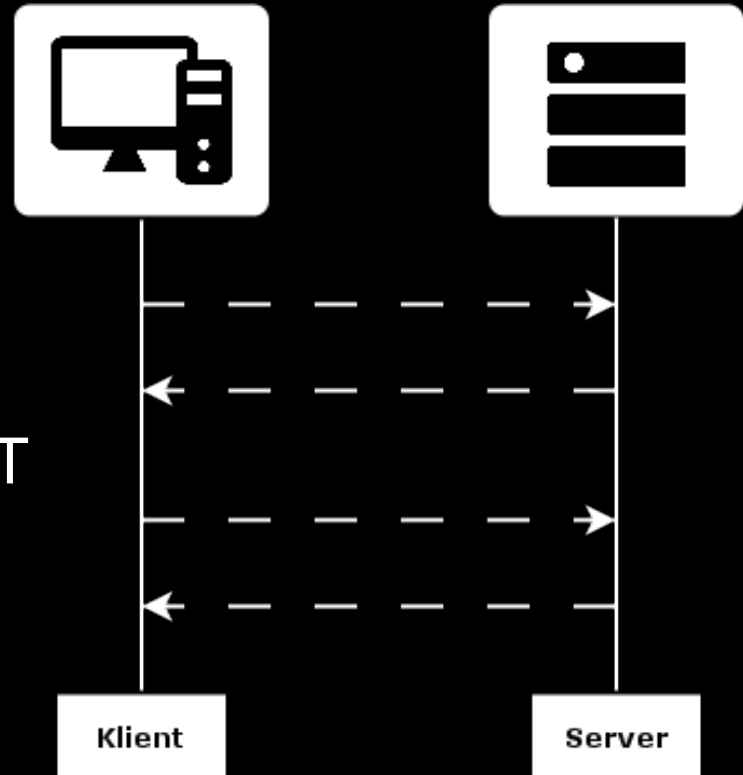
Jak se posílají data ?

# REST



# Representational State Transfer

- bezstavový
- orientace na data
- url reprezentuje “zdroj”
- HTTP metody
  - GET, POST, DELETE, PUT
- JSON



# Representational State Transfer

- bezstavový
- orientace na data
- url reprezentuje “zdroj”
- HTTP metody
  - GET, POST, DELETE, PUT
- JSON

```
{  
  "id": 90,  
  "name": "Janáček",  
  "degree": "10",  
  "alcohol": 3.46,  
  "brewery": {  
    "id": 19,  
    "name": "Uherský Brod",  
    "year": 1894  
  }  
}
```



# Representational State Transfer

URL zdroje - /api/v1/brewery

GET                /api/v1/brewery

POST              /api/v1/brewery

# Representational State Transfer

URL zdroje - `/api/v1/brewery/42`

GET `/api/v1/brewery/42`

PUT `/api/v1/brewery/42`

DELETE `/api/v1/brewery/42`

# Representational State Transfer

URL zdroje umožňuje zanoření

`/api/v1/brewery/42/beer/12/pub/....`

URL může obsahovat parametry

`/api/v1/brewery?page=2&limit=20`

# REST v AngularJS - ngResource

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

```
var list = Brewery.query({'page' : 2});
```

```
var oldBrewery = Brewery.get({id : 42});
```

```
oldBrewery.$delete();
```

```
var brewery = new Brewery();
```

```
brewery.name = 'Janáček';
```

```
brewery.$save();
```

**Autentizace & Autorizace**

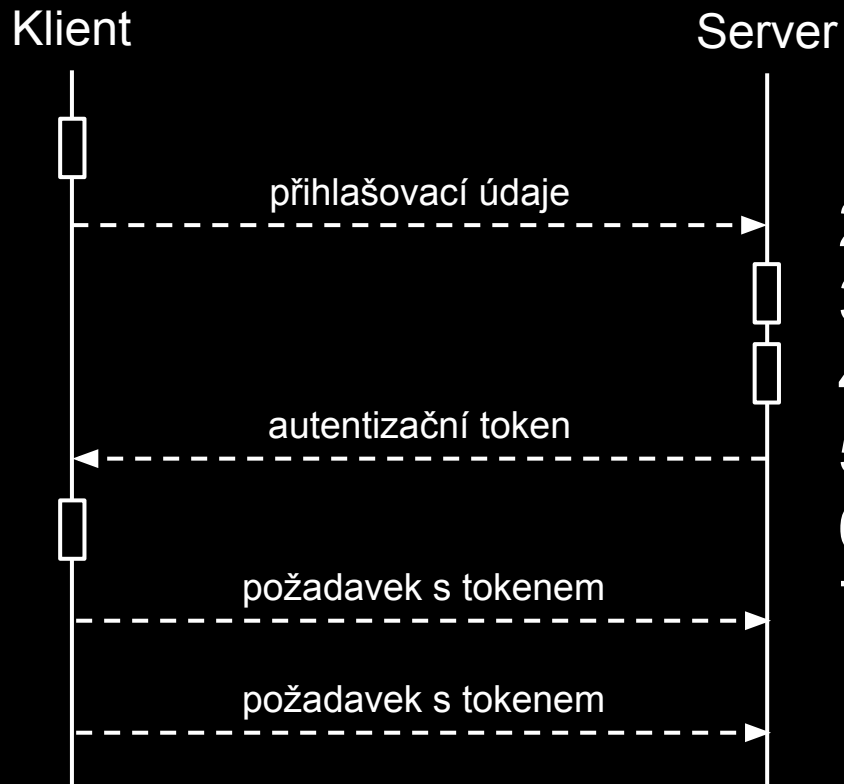
# Autentizace

- ověření identity uživatele

# Autorizace

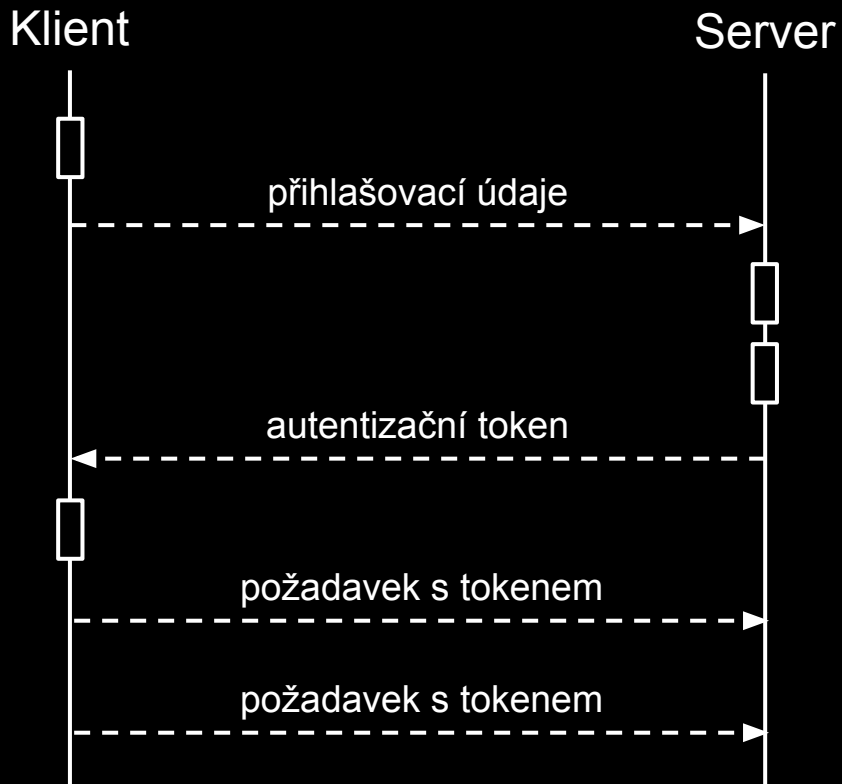
- zabezpečení přístupu ke chráněným zdrojům

# Autentizace



1. klient zadá jméno a heslo
2. odešle je na server
3. server ověří uživatele
4. vytvoří autentizační token
5. pošle ho klientovi
6. klient si jej poznamená
7. přikládá k požadavkům

# Autentizace - token



## Co je to ten TOKEN?

- "přístupová kartička"

## Proč se používá

- bezpečnost
- bezestavovost
- decentralizace
- JWT



# Autorizace

## Na straně serveru

- zabezpečení REST
- stavové kódy HTTP
  - 401 - Unauthorized
  - 403 - Forbidden

## Na straně klienta

- skrýt prvky GUI
- zabránit v přístupu
- reagovat na stavové kódy HTTP

# Autentizace & autorizace

Co se stává, když vyprší platnost přihlášení ?

Co by si asi uživatel přál aby se stalo ?

Jak toho dosáhneme ?

Správnou obsluhou  
stavových kódů HTTP

# Co nám umožní takovéto chování?

## Promise

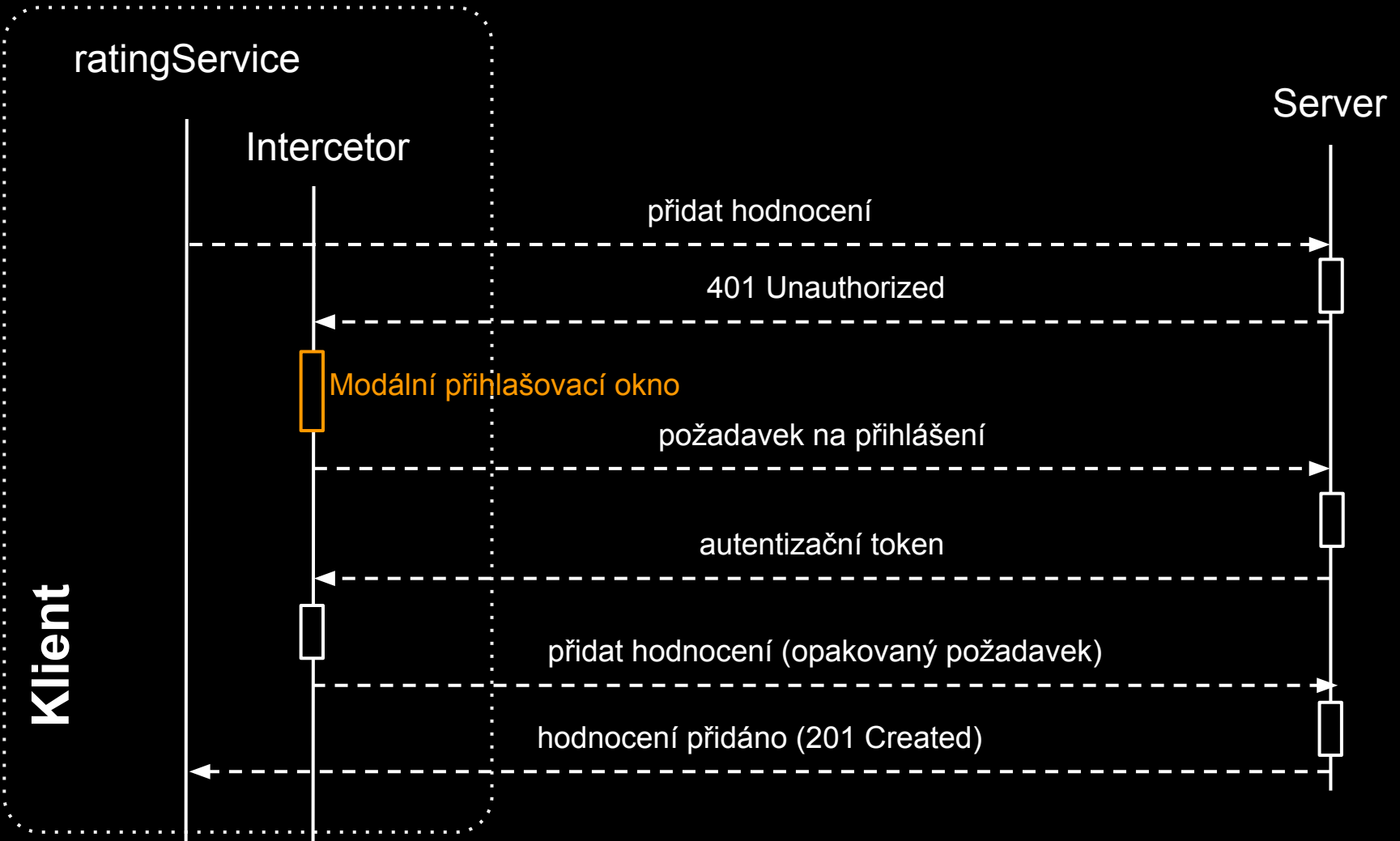
- objekt reprezentující (budoucí) výsledek asynchronní operace

# Asynchroni operace - callback

```
step1(function (value1) {  
    step2(value1, function(value2) {  
        step3(value2, function(value3) {  
            step4(value3, function(value4) {  
                // Do something with value4  
            });  
        });  
    });  
});
```

# Asynchroni operace - promise

```
functionWithPromise()  
  .then(promisedStep2)  
  .then(promisedStep3)  
  .then(promisedStep4)  
  .then(function (value4) {  
    // Do something with value4  
  })  
  .catch(function (error) {  
    // Handle any error from all above steps  
  })
```



# Json Web Token

- **zažívá poměrně velký HYPE**
- **jednoduchá implementace**
  - **server jej vydá a ověří**
  - **klient se chová jako k jinému tokenu**
- **bezstavový token**
  - **server neví že jej vydal**
  - **odhlášení = zahození tokenu**
- **může nést informace**
  - **mohou obsahovat dobu platnosti**

Devstack

KARMA

node JS





# Jak se takováto aplikace nasazuje

Na samostatný http server

- html, javascript, css
- frontend - **angular.cz**
- api - **api.angular.cz**

Spolu se aplikací, která poskytuje api

- frontend - **angular.cz**
- api - **angular.cz/api**

# Začlenění do java aplikace

Jak vypadá webová aplikace v javě

- aplikační server
- war archiv
- assety ve webapp / static ...
- sestavení aplikace - maven

# Sestavení aplikace - Maven

beer-app

- beer-app-frontend
- beer-app-webapp

- maven-exec-plugin
- maven-frontend-plugin
  - install-node-and-npm
  - npm
  - bower
  - gulp
  - karma

# Co si odnést?

- prohlížeč je “platforma” na které se dá napsat slušná aplikace a nemusí to být trápení pro vývojáře, ani pro uživatele
- když píšete webovou aplikaci, pište ji s vědomím, že běží v browseru, ne s myšlenkou, jak uživateli omezit nativní funkce prohlížeče

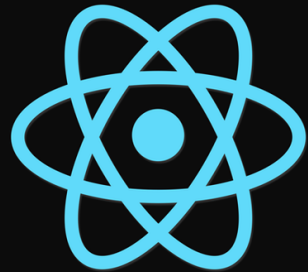
# AngularJS

není náhrada za jQuery, ale plnohodnotný framework pro psaní aplikací běžících v prohlížeči

- DI, testování
- dokumentace a velká komunita
- nástroje

používá Seznam, Netflix, Vevo, PS3 - youtube...

# Alternativy



**React**   
Flux Architecture

[todomvc.com](http://todomvc.com)

# Jak začít?

[jsfiddle.net](https://jsfiddle.net) - [plnkr.co](https://plnkr.co) - [jsbin.com](https://jsbin.com) - [codepen.io](https://codepen.io)

[angularjs.org](https://angularjs.org)

[codio.com](https://codio.com)

**Zdroje - [www.angular.cz/jug-24-03-2015](http://www.angular.cz/jug-24-03-2015)**

