



webnextconf.eu

Create a Single Page web Application with AngularJS

Antonio Di Motta

<http://www.dimotta.net>

<http://twitter.com/dimotta>

<http://github.com/antdimot>

Who I am?

I'm Antonio Di Motta and I'm working for Mate Consulting in Salerno as IT Project Engineer (PE).

I'm accountable to design, develop and deliver some projects about industry and transport sectors.

Yes, I'm the man that usually:

- 1) Goes to the customer to understand what he wants (the hard part).
- 2) Getting the requirements
- 3) Designs the solution (the cheapest possible...)
- 4) Implements the solution with my team (as soon as possible)
- 5) Goes to the customer for delivering the solution
- 6) Checks that the integration tests will be passed
- 7) Trains customer staff

Agenda

- Key concepts of Single Page Application
- AngularJS features
- Demos
- An open source project based on AngularJS

Single Page Application (SPA)

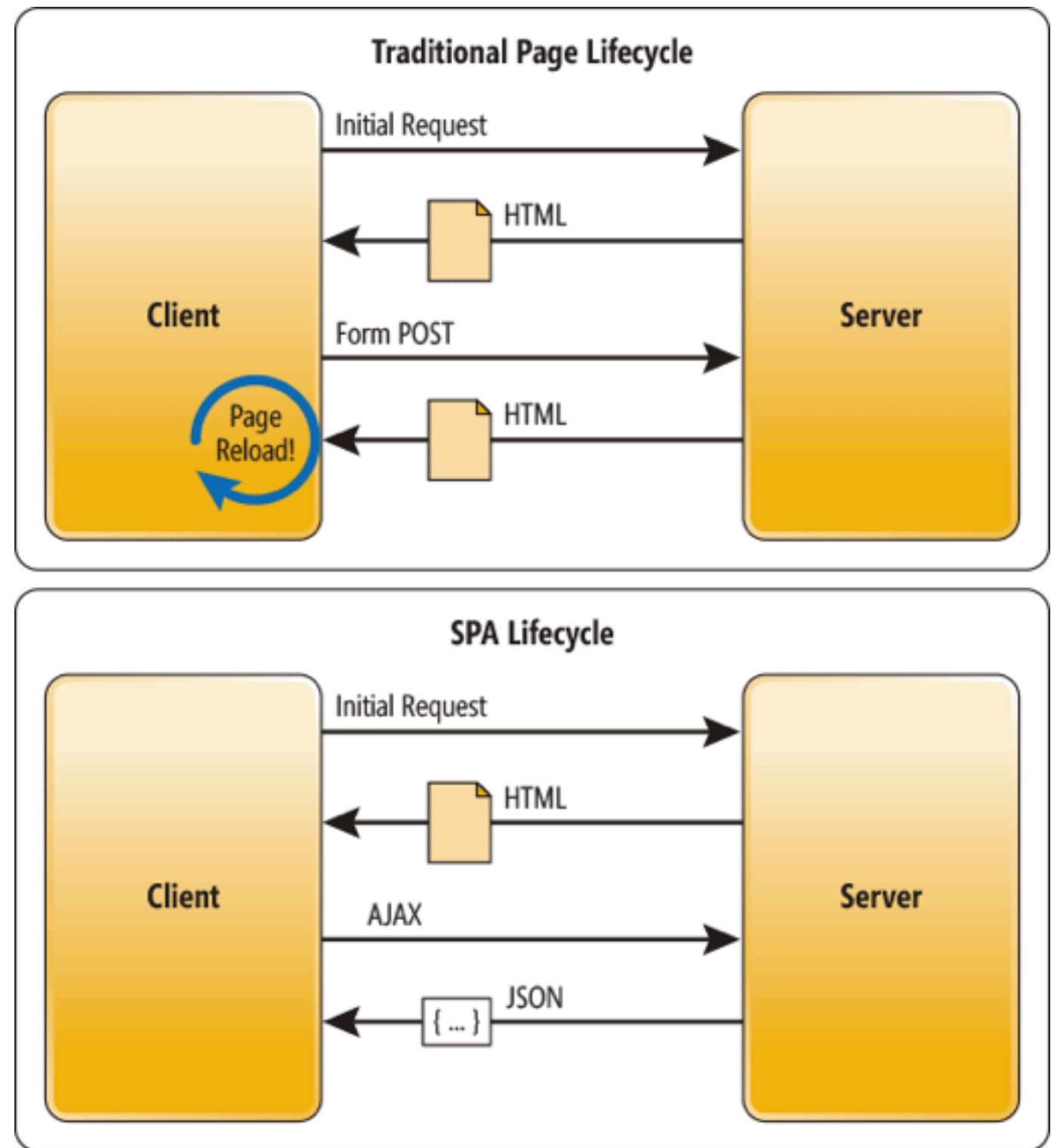
SPA's popularity has been increasing in recent times because from the start, everything is done from the **front-end** of a web application using **Ajax** (to interacting with the server), **HTML templates**, a good **MVC/MVVM** framework, and of course, a lot of **JavaScript**.

This kind of web application fits on a **single web page** and **able to update parts** of an interface without sending or receiving a full-page request.

The goal is to provide a more **fluid user experience** and a **rich interface**.

Page Lifecycle

Traditional web applications reside mostly on the server, with heavy traffic flow between browser and server using GETs, POSTs (and in webforms, the dreaded postback) being the order of the day.

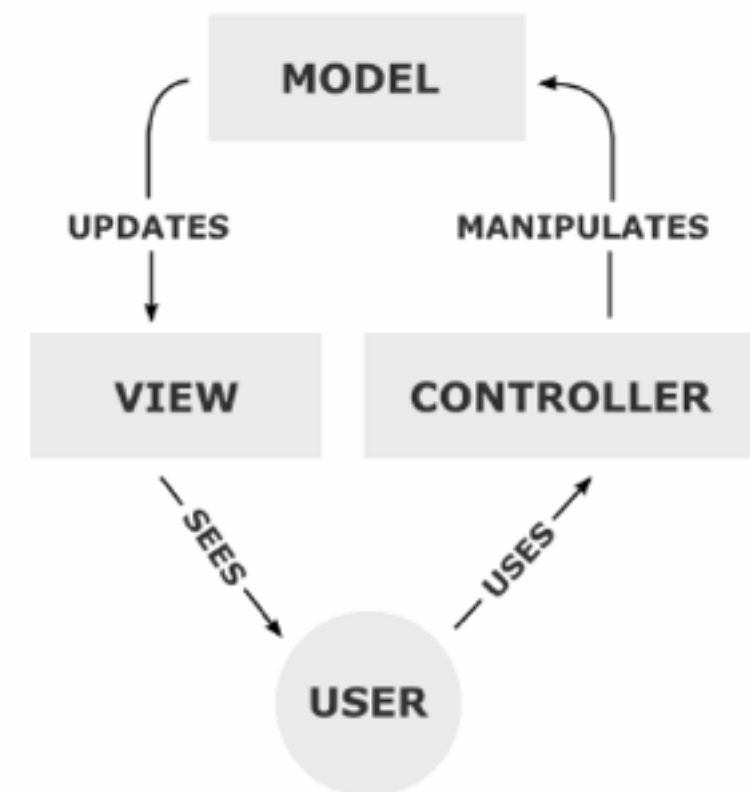


Understanding the MVC pattern

Many web frameworks that we have today follows the standard MVC pattern.

Let's review the concepts of MVC:

- A **controller** can send commands to the model to update the model's state. It can also send commands to its associated view to change the view's presentation of the model
- A **model** stores data that is retrieved according to commands from the controller and displayed in the view.
- A **view** generates an output presentation to the user based on changes in model.



AngularJS is a full-featured SPA framework

It supports both types of binding one-way and two-way data bindings

It supports MVC pattern

It supports static template and angular templates

You can add custom directives

It supports REST full services

It supports form validations

It supports both client and server communication

It supports dependency injection

Event Handlers

What are the main features in AngularJS?

Scope

Model

Data binding

Controller

View

Templates

Filters

Directives

jqLite

Dependency Injection

Services

Validation

MVC

Routing

Getting started with AngularJS



<http://angularjs.org>

```
mactony:Examples antdimot$ bower install angular --save
bower angular#*                cached git://github.com/angular/bower-angular.git#1.4.6
bower angular#*                validate 1.4.6 against git://github.com/angular/bower-angular.git#*
mactony:Examples antdimot$
```

It is time to show the code.

Four ways to organize the code

When starting a new AngularJS project a common question is "how do we organize the code?".

There are many ways, tendencies, and techniques to organize the project's code within files and directories. Specially with projects based on AngularJS is possible summarize all techniques with four styles:

- 1 The inline style
- 2 The stereotyped style
- 3 The specific style
- 4 The domain style

The inline style

It's useful for making a presentation, for developing a fast and disposable application prototype or to evaluate a potential product idea.

The only project structure consists of index.html file with inline declarations for the scripts and style.

app/	-> files of the application
index.html	-> main html file
angular.js	-> AngularJS script

The stereotyped style

This style is useful for small apps with a limited number of components such as controllers, services, directives, and filters.

app/	-> files of the application
lib/	-> javascript libraries
angular.js	-> AngularJS script
css/	-> css files
app.css	-> application stylesheet
js/	-> application javascript components
config.js	-> main application script
controllers.js	-> all controllers script
services.js	-> all services script
directives.js	-> all directives script
partials/	-> partial view directory
blog.html	-> blog view
post.html	-> post view
comment.html	-> comment view
index.html	-> main html file

The specific style

When the project is growing can be useful splitting the scripts into specific ones.

app/	-> files of the application
lib/	-> javascript libraries
angular.js	-> AngularJS script
css/	-> css files
app.css	-> application stylesheet
js/	-> javascript application components
controllers/	-> controllers directory
blogCtrl.js	-> blog controller
postCtrl.js	-> post controller
commentCtrl.js	-> comment controller
services/	-> services directory
directives/	-> directives directory
config.js	-> main application script
partials/	-> partial view directory
blog.html	-> blog view
post.html	-> post view
comment.html	-> comment view
index.html	-> main html file

The domain style

app/	-> files of the application
lib/	-> javascript libraries
angular.js	-> AngularJS script
application/	-> application module directory
services/	-> services directory
directives/	-> directives directory
app.css	-> application stylesheet
config.js	-> main application script
blog/	-> blog module directory
blogCtrl.js	-> blog controller
blog.css	-> blog stylesheet
blog.html	-> blog view
post/	-> post directory
postCtrl.js	-> post controller
post.css	-> post stylesheet
post.html	-> post view
comment/	-> comment directory
commentCtrl.js	-> comment controller
comment.css	-> comment stylesheet
comment.html	-> comment view
index.html	-> main html file

Chirp (<https://github.com/antdimot/chirp>)

A simple **Mean** application that implements some features of Twitter:

- show public timeline
- user authentication
- show home timeline
- post a message (chirp)
- list of followers
- list of following

MEAN, the full stack javascript

M stands for **MongoDB**, the world's leading **NoSQL** database. That's a kind of document type database that stores its data into a JSON-like formatted binary file called BSON (Binary JSON).

E stands for **Express**, a lightweight, minimalist framework built for Node.js. It's been created for web applications and APIs.

A stands for **AngularJS**, it's a client-side framework for MVC/MVVM done in JavaScript. The standard adopted is the closest to the MVVM pattern and is very robust and highly suitable for SPA.

N stands for **Node.js**, the foundation of Express. It runs on Chrome's V8 engine and is capable of non-blocking, event-driven I/O.

Chirp architecture

AngularJS

SPA

NodeJS

Server

Express

rest api

Mongodb

Database



Sponsors

All of this wouldn't be possible without the support of these awesome partners



Microsoft

BrainTree_Dev.

A PayPal Company



scientiamobile



Partners



REFINED



webnextconf.eu