

ANGULAR

Data binding

Con data binding ci si riferisce alla sincronizzazione automatica che avviene tra model e view. In angular questo avviene in maniera bidirezionale, ogni volta che il modello viene modificato viene aggiornata la vista ed ogni volta che viene modificata la view (attraverso input / interazione dell'utente) viene aggiornato automaticamente il model.

<https://docs.angularjs.org/guide/databinding>

<https://docs.angularjs.org/guide/scope>

Controllers

Sono definiti come funzioni javascript utilizzate come costruttore. I controller possono essere utilizzati direttamente all'interno del DOM oppure in associazione a una route oppure associato a una direttiva. La funzione "controller" viene utilizzata per inizializzare il model e per esporre le funzionalità che si desidera fornire all'utente.

<https://docs.angularjs.org/guide/controller>

Services / Factory / Provider

Sono oggetti javascript. Vengono utilizzati per definire parti dell'applicazione che possono essere riutilizzati all'interno dell'applicazione. I servizi possono essere iniettati all'interno dei controller o di altri servizi.

<https://docs.angularjs.org/guide/service>

<https://docs.angularjs.org/guide/di>

<https://docs.angularjs.org/guide/providers>

Filters

Sono utilizzati per formattare o applicare una trasformazione ai dati. Possono essere utilizzati direttamente all'interno del DOM oppure all'interno di controller e servizi utilizzando il servizio \$filter.

<https://docs.angularjs.org/guide/filter>

Directives

Sono utilizzate per definire veri e propri componenti all'interno della propria applicazione, andando ad associare particolari funzionalità (e template) ad attributi od elementi HTML.

<https://docs.angularjs.org/guide/directive>

Routing

All'interno di angular è possibile associare a delle url un controller e un template specifici. Per poter utilizzare le funzionalità di routing va aggiunto il modulo "angular-route" come dipendenza della propria applicazione.

<https://docs.angularjs.org/api/ngRoute>

Development Tools

GIT

Sistema di versionamento del codice.

<https://git-scm.com/>

NPM

Gestore di pacchetti per NodeJS. Viene utilizzato per gestire installare tutti i tools necessari per lo sviluppo.

Alcuni esempi di tools utilizzati:

- Bower
- Grunt
- Plugin per grunt
- Lite server

Le dipendenze vengono lette dal file "package.json"

<https://www.npmjs.com>

BOWER

Gestore di pacchetti per il web. Viene utilizzato per gestire le dipendenze dell'applicazione.

Le dipendenze vengono lette dal file "bower.json". Eventuali opzioni di configurazione aggiuntive devono essere salvate in un file ".bowerrc".

<http://bower.io/>

GRUNT

The javascript task runner. Viene utilizzato per automatizzare i processi di compilazione e test dell'applicazione. Sono disponibili molti plugin per eseguire i task più comuni, i plugin vengono installati utilizzando NPM. Alcuni esempi sono:

- Compilazione sass/compass/less
- Minificazione di javascript e css
- Inserimento js e css nella index.html
- Esecuzione unit testing

I task vengono definiti nel file "Gruntfile.js".

<http://gruntjs.com/>

Startup dell'applicazione

1. Installare NPM, per installare NPM è necessario installare nodejs. Al seguente link è possibile scaricare l'installer <https://nodejs.org/en/download/>
2. Opzionale: Installare GIT. Al seguente link è possibile scaricare l'installer <https://git-scm.com/downloads>.
3. Clonare il progetto angular con la seguente istruzione: "git clone <https://github.com/Coolshop/angular-start.git>". In alternativa si può scaricare uno zip contenente i file al seguente link <https://github.com/Coolshop/angular-start>
4. Installare i tool di sviluppo necessari per lo sviluppo eseguendo all'interno della cartella principale dell'applicazione la seguente istruzione "npm install".
5. Scaricare le dipendenze dell'applicazione sviluppo eseguendo all'interno della cartella principale dell'applicazione la seguente istruzione "bower install".
6. "Compilare" l'applicazione con grunt eseguendo all'interno della cartella principale dell'applicazione la seguente istruzione "grunt develop".
7. Eseguire "grunt developWatch" per avviare la compilazione automatica dell'applicazione alla modifica dei file.
8. Avviare l'applicazione eseguendo all'interno della cartella principale dell'applicazione la seguente istruzione "npm start". Questo comando lancerà un web server standalone.