

ionic

Miguel Navarro Cabrero

Prerrequisitos

- GitHub

```
$ sudo apt-get install git
```

- Node.js y npm

<https://nodejs.org/es/download/package-manager/>

Instalación

```
$ npm install -g cordova ionic
```

- Plugins

<http://ngcordova.com/docs/plugins/>

Preparando servidor Node

```
$ git clone https://github.com/ccoenraets/ionic-tutorial
```

```
$ cd ionic-tutorial/server
```

```
$ npm install
```

```
$ node server
```

- Comprobar servicio REST:
<http://localhost:5000/sessions>
<http://localhost:5000/sessions/1>

Creación de la aplicación

```
$ git clone https://github.com/navarromiguel/ionic-tutorial.git
```

```
$ ionic start conference sidemenu
```

```
$ cd conference
```

- Arrancar la aplicación:

```
$ ionic serve
```

```
$ ionic serve -l
```

Creación de un servicio (I)

- Vamos a crear un servicio para la **Session**. En *conference/www/js* crear un archivo llamado *services.js* y definimos dentro el siguiente módulo:
`angular.module('starter.services', ['ngResource'])`
- A continuación definimos el servicio asociándolo con el módulo creado:
`.factory('Session', function ($resource) {
 return
 $resource('http://localhost:5000/sessions/:sessionId');
});`

Creación de un servicio (II)

- En *index.html*, después de *ionic-bundle.js* incluimos la dependencia para **\$resource**:

```
<script  
src="lib/ionic/js/angular/angular-resource.min.js"></script>
```
- Después de *app.js* incluimos el *services.js* que acabamos de crear:

```
<script src="js/services.js"></script>
```

Creación de controladores para Session

```
$ git checkout -f paso1
```

- En *js/controllers.js* inyectamos el módulo que hemos creado para poder hacer uso de él.
- Reemplazamos el **PlayListsCtrl** por un nuevo controlador para las **Sessions**:

```
.controller('SessionsCtrl', function($scope, Session) {  
    $scope.sessions = Session.query();  
})
```

- Reemplaza **PlayListCtrl** por el controlador para los detalles de una **Session**:
- ```
.controller('SessionCtrl', function($scope, $stateParams, Session) {
 $scope.session = Session.get({sessionId: $stateParams.sessionId});});
```



# Creando templates (I)

- Dentro del directorio *conference/www/templates* renombramos *playlists.html* y lo llamamos **sessions.html** y lo implementamos tal que así:

```
<ion-view view-title="Sessions">
<ion-content><ion-list>
<ion-item ng-repeat="session in sessions"
href="#/app/sessions/{{session.id}}"
 {{session.title}}
</ion-item></ion-list></ion-content>
</ion-view>
```

## Creando templates (II)

- Renombra *playlist.html* por **session.html** e impleméntalo de la siguiente forma:

```
<ion-view view-title="Session"><ion-content><div class="list card"><div class="item"><h3>{{session.time}}</h3><h2>{{session.title}}</h2><p>{{session.speaker}}</p></div><div class="item item-body"><p>{{session.description}}</p></div><div class="item tabs tabs-secondary tabs-icon-left"><i class="icon ion-thumbsup"></i>Like<i class="icon ion-chatbox"></i>Comment<i class="icon ion-share"></i>Share</div></div></ion-content></ion-view>
```

# Routing (I)

- En *js/app.js* reemplaza el state **app.playlists** por el siguiente:

```
.state('app.sessions', {
 url: "/sessions",
 views: {
 'menuContent': {
 templateUrl: "templates/sessions.html",
 controller: 'SessionsCtrl'
 }
 }
})
```

## Routing (II)

- Reemplaza el **app.single** por el siguiente state:

```
.state('app.session', {
 url: "/sessions/:sessionId",
 views: {
 'menuContent': {
 templateUrl: "templates/session.html",
 controller: 'SessionCtrl'
 }
 }
});
```

## Routing (III)

- Modifica la ruta por defecto para que apunte a */app/sessions*.
- En menu.html modifica la etiqueta y el href del item **Playlists** para que nos lleve al listado de sessions.
- Comprueba que todo funciona correctamente.

# Building

- iOS

```
$ ionic platform add ios
```

```
$ ionic build ios
```

- Android

```
$ ionic platform add android
```

```
$ ionic build android
```

```
$ ionic run android
```

```
$ ionic emulate android
```

# Integración con Facebook (I)

- Crear una app de Facebook:  
<https://developers.facebook.com/apps>
- Seleccionar **www** como plataforma
- En **OAuth Settings**, añadir en **Valid OAuth redirect URIs** las siguientes:  
<http://localhost:8100/oauthcallback.html> (acceso vía ionic serve)  
[https://www.facebook.com/connect/login\\_success.html](https://www.facebook.com/connect/login_success.html) (acceso vía Cordova)

# Integración con Facebook (II)

- Copia **openfb.js** y **ngopenfb.js** de *ionic-tutorial/resources* a *conference/www/js*.
- Copia **oauthcallback.html** y **logoutcallback.html** de *ionic-tutorial/resources* a *conference/www*.
- Incluye en *index.html* antes de *app.js*:  

```
<script src="js/openfb.js"></script>
<script src="js/ngopenfb.js"></script>
```
- Añade la dependencia **ngOpenFB** al módulo starter en *app.js*.



# Integración con Facebook (III)

- Inyecta **ngFB** en la función **run()** e inicialízala en la primera línea:  
`ngFB.init({appId: 'MI_APP_ID'});`
- Añade **ngOpenFB** en el módulo *starter.controllers* e inyecta **ngFB** en *AppCtrl*.

# Integración con Facebook (IV)

Implementa la función login:

```
$scope.fbLogin = function () {
 ngFB.login({scope: 'email,read_stream,publish_actions'}).then(
 function (response) {
 if (response.status === 'connected') {
 console.log('Facebook login succeeded');
 $scope.closeLogin();
 } else {
 alert('Facebook login failed');
 }
 })
 });
};
```

# Integración con Facebook (V)

- En *login.html* agrega un nuevo botón para login con Facebook:

```
<label class="item">
 <button class="button button-block button-positive">
 Login with Facebook
 </button>
</label>
```

- Implementa el evento para cuando se haga click se invoque a **fbLogin()**.
- Testea la aplicación y comprueba que todo funciona correctamente.

# Integración con Facebook (VI)

- Crea un controlador para el perfil del usuario:

```
.controller('ProfileCtrl', function ($scope, ngFB) {
 ngFB.api({
 path: '/me',
 params: {fields: 'id,name'}
 }).then(
 function (user) {
 $scope.user = user;
 },
 function (error) {
 alert('Facebook error: ' + error.error_description);
 });
});
```

# Integración con Facebook (VII)

- Crea un nuevo template llamado profile.html con el siguiente contenido:

```
<ion-view view-title="Profile"><ion-content class="has-header"><div
class="list card">
 <div class="item">
 <h2>{{user.name}}</h2>
 <p>{{user.city}}</p></div><div class="item item-body">
</div></div>
 </ion-content>
</ion-view>
```

# Integración con Facebook (VIII)

- Añade una ruta para el perfil del usuario con url */profile*
- Añade un nuevo item de menú para ir al perfil del usuario.
- Inyecta **ngFB** en *SessionCtrl* y añade la siguiente función para compartir:  

```
$scope.share = function (event) {ngFB.api({method: 'POST',path:
'/me/feed',params: {
 message: "I'll be attending: '" + $scope.session.title + "' by
" +
 $scope.session.speaker}}).then(function () {alert('The session
was shared on Facebook');},function () {alert('An error occurred while
sharing this session on Facebook');}});};
```

# Integración con Facebook (IX)

- Añade una invocación a la función *share()* desde `session.html`
- Testea la aplicación.

```
$ git checkout -f paso3
```