LECTURE 4

NAVIGATION, LOCAL STORAGE

DANIEL RYS
JAN VÁCLAVÍK

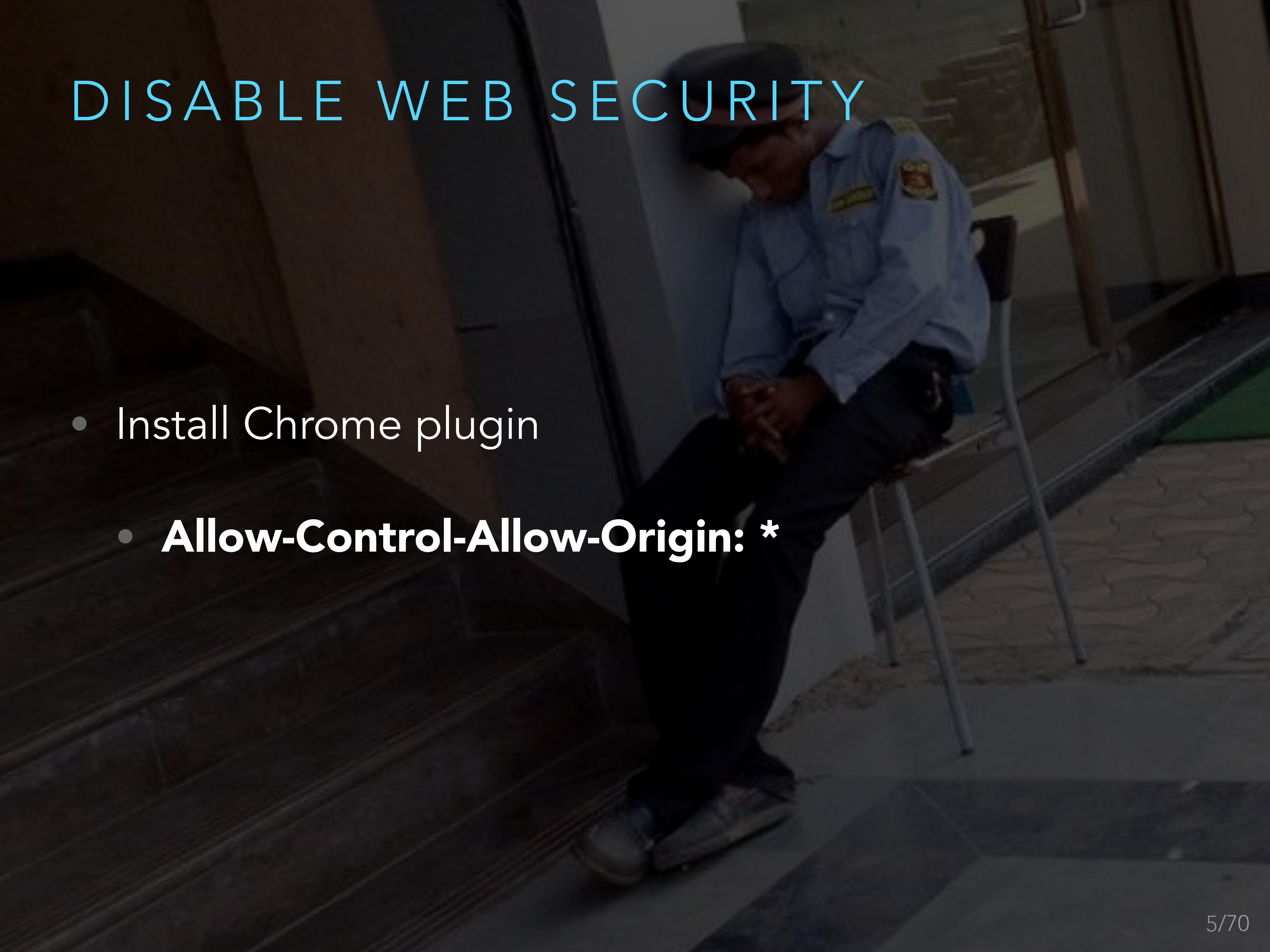
OVERVIEW

- Implement movie detail
- Get data from API
- Mark movie as watched (in Local Storage)

CLONE AND INSTALL REPOSITORY

```
$ git clone https://github.com/jvaclavik/codecamp-movies-app movie-app-api
$ cd movie-app-api/
$ npm install
$ ionic serve
```

GET DATA FROM API



DISABLE WEB SECURITY

- Install Chrome plugin
  - **Allow-Control-Allow-Origin: ***

USING THEMOVIEDB.ORG

Documentation: docs.themoviedb.apiary.io

INSTALL POSTMAN

http://getpostman.com

GET DATA FROM API

www/app/config.js

```
app.constant("APP_CONFIG", {
    debugMode: true,
    storagePrefix: "test_",
    defaultState: "/menu/home",

    apiKey: "4aa883f95999ec813b8bfaf319f3972b",
    apiUrl: "api.themoviedb.org/3/",
    apiEndpoints: {
        moviesPopular: "movie/popular"
    },

    getApiUrl: function (endpoint){
        return "http://" + this.apiUrl + this.apiEndpoints[endpoint]
            + "?api_key="+ this.apiKey;
    }
});
```

GET DATA FROM API

www/app/config.js

Copy this key

```
app.constant("APP_CONFIG", {
    debugMode: true,
    storagePrefix: "test_",
    defaultState: "/menu/home",

    apiKey: "4aa883f95999ec813b8bfaf319f3972b",
    apiUrl: "api.themoviedb.org/3/",
    apiEndpoints: {
        moviesPopular: "movie/popular"
    },

    getApiUrl: function (endpoint){
        return "http://" + this.apiUrl + this.apiEndpoints[endpoint]
            + "?api_key="+ this.apiKey;
    }
});
```

GET DATA FROM API

www/app/config.js

```
app.constant("APP_CONFIG", {
    debugMode: true,
    storagePrefix: "test_",
    defaultState: "/menu/home",

    apiKey: "4aa883f95999ec813b8bfaf319f3972b",
    apiUrl: "api.themoviedb.org/3/",
    apiEndpoints: {
        moviesPopular: "movie/popular"
    },

    getApiUrl: function (endpoint){
        return "http://" + this.apiUrl + this.apiEndpoints[endpoint]
            + "?api_key="+ this.apiKey;
    }
});
```

API URL

GET DATA FROM API

www/app/config.js

```
app.constant("APP_CONFIG", {
    debugMode: true,
    storagePrefix: "test_",
    defaultState: "/menu/home",

    apiKey: "4aa883f95999ec813b8bfaf319f3972b",
    apiUrl: "api.themoviedb.org/3/",
    apiEndpoints: {
        moviesPopular: "movie/popular"
    },

    getApiUrl: function (endpoint){
        return "http://" + this.apiUrl + this.apiEndpoints[endpoint]
            + "?api_key="+ this.apiKey;
    }
});
```

API endpoint for
popular movies

GET DATA FROM API

www/app/config.js

```
app.constant("APP_CONFIG", {
    debugMode: true,
    storagePrefix: "test_",
    defaultState: "/menu/home",

    apiKey: "4aa883f95999ec813b8bfaf319f3972b",
    apiUrl: "api.themoviedb.org/3/",
    apiEndpoints: {
        moviesPopular: "movie/popular"
    },

    getApiUrl: function (endpoint){
        return "http://" + this.apiUrl + this.apiEndpoints[endpoint]
            + "?api_key="+ this.apiKey;
    }
});
```

Composing API URL + endpoint + key

MOVIE SERVICE

www/model/movies-service.js

```
app.factory('MoviesService', function ($http, APP_CONFIG) {
    return new (function () {
        var service = this;
        service.data = {};
        service.getMovies = function () {
            var req = {
                method: "GET",
                url: APP_CONFIG.getApiUrl("moviesPopular")
            };
            $http(req).success(function (response) {
                service.data.movies = response.results;
            }).error(function (data, status, headers, config) {
                console.error("error");
            });
        };
        service.getMovieById = function (id) {
            //TODO
        };
        service.getMovies();
    })();
});
```

MOVIE SERVICE

www/model/movies-service.js

```
app.factory('MoviesService', function ($http, APP_CONFIG) {
    return new (function () {
        var service = this;
        service.data = {};
        service.getMovies = function () {
            var req = {
                method: "GET",
                url: APP_CONFIG.getApiUrl("moviesPopular")
            };
            $http(req).success(function (response) {
                service.data.movies = response.results;
            }).error(function (data, status, headers, config) {
                console.error("error");
            });
        };
        service.getMovieById = function (id) {
            //TODO
        };
        service.getMovies();
    })();
});
```

Method for getting popular movies

MOVIE SERVICE

www/model/movies-service.js

```
app.factory('MoviesService', function ($http, APP_CONFIG) {
    return new (function () {
        var service = this;
        service.data = {};
        service.getMovies = function () {
            var req = {
                method: "GET",
                url: APP_CONFIG.getApiUrl("moviesPopular")
            };
            $http(req).success(function (response) {
                service.data.movies = response.results;
            }).error(function (data, status, headers, config) {
                console.error("error");
            });
        };
        service.getMovieById = function (id) {
            //TODO
        };
        service.getMovies();
    })();
});
```

Request object which defines method, URL, headers etc.

MOVIE SERVICE

www/model/movies-service.js

```
app.factory('MoviesService', function ($http, APP_CONFIG) {
    return new (function () {
        var service = this;
        service.data = {};
        service.getMovies = function () {
            var req = {
                method: "GET",
                url: APP_CONFIG.getApiUrl("moviesPopular")
            };
            $http(req).success(function (response) {
                service.data.movies = response.results;
            }).error(function (data, status, headers, config) {
                console.error("error");
            });
        };
        service.getMovieById = function (id) {
            //TODO
        };
        service.getMovies();
    })();
});
```

We use GET method for getting data

MOVIE SERVICE

www/model/movies-service.js

```javascript
app.factory('MoviesService', function ($http, APP_CONFIG) {
    return new (function () {
        var service = this;
        service.data = {};
        service.getMovies = function () {
            var req = {
                method: "GET",
                url: APP_CONFIG.getApiUrl("moviesPopular")
            };
            $http(req).success(function (response) {
                service.data.movies = response.results;
            }).error(function (data, status, headers, config) {
                console.error("error");
            });
        };
        service.getMovieById = function (id) {
            //TODO
        };
        service.getMovies();
    })();
});
```

Function for composing API request URL

MOVIE SERVICE

www/model/movies-service.js

```
app.factory('MoviesService', function ($http, APP_CONFIG) {
    return new (function () {
        var service = this;
        service.data = {};
        service.getMovies = function () {
            var req = {
                method: "GET",
                url: APP_CONFIG.getApiUrl("moviesPopular")
            };
            $http(req).success(function (response) {
                service.data.movies = response.results;
            }).error(function (data, status, headers, config) {
                console.error("error");
            });
        };
        service.getMovieById = function (id) {
            //TODO
        };
        service.getMovies();
    })();
});
```

HTTP request

MOVIE SERVICE

Dependency injection

www/model/movies-service.js

```
app.factory('MoviesService', function ($http, APP_CONFIG) {
    return new (function () {
        var service = this;
        service.data = {};
        service.getMovies = function () {
            var req = {
                method: "GET",
                url: APP_CONFIG.getApiUrl("moviesPopular")
            };
            $http(req).success(function (response) {
                service.data.movies = response.results;
            }).error(function (data, status, headers, config) {
                console.error("error");
            });
        };
        service.getMovieById = function (id) {
            //TODO
        };
        service.getMovies();
    })();
});
```

HTTP request

MOVIE SERVICE

www/model/movies-service.js

```
app.factory('MoviesService', function ($http, APP_CONFIG) {
    return new (function () {
        var service = this;
        service.data = {};
        service.getMovies = function () {
            var req = {
                method: "GET",
                url: APP_CONFIG.getApiUrl("moviesPopular")
            };
            $http(req).success(function (response) {
                service.data.movies = response.results;
            }).error(function (data, status, headers, config) {
                console.error("error");
            });
        };
        service.getMovieById = function (id) {
            //TODO
        };
        service.getMovies();
    })();
});
```

Success callback

Defines what happens after getting data

MOVIE SERVICE

www/model/movies-service.js

```
app.factory('MoviesService', function ($http, APP_CONFIG) {
    return new (function () {
        var service = this;
        service.data = {};
        service.getMovies = function () {
            var req = {
                method: "GET",
                url: APP_CONFIG.getApiUrl("moviesPopular")
            };
            $http(req).success(function (response) {
                service.data.movies = response.results;
            }).error(function (data, status, headers, config) {
                console.error("error");
            });
        };
        service.getMovieById = function (id) {
            //TODO
        };
        service.getMovies();
    })();
});
```

“response” is what we get from server

MOVIE SERVICE

www/model/movies-service.js

```
app.factory('MoviesService', function ($http, APP_CONFIG) {
    return new (function () {
        var service = this;
        service.data = {};
        service.getMovies = function () {
            var req = {
                method: "GET",
                url: APP_CONFIG.getApiUrl("moviesPopular")
            };
            $http(req).success(function (response) {
                service.data.movies = response.results;
            }).error(function (data, status, headers, config) {
                console.error("error");
            });
        };
        service.getMovieById = function (id) {
            //TODO
        };
        service.getMovies();
    })();
});
```

Save "results" to our "data.movies"

MOVIE SERVICE

www/model/movies-service.js

```
app.factory('MoviesService', function ($http, APP_CONFIG) {
    return new (function () {
        var service = this;
        service.data = {};
        service.getMovies = function () {
            var req = {
                method: "GET",
                url: APP_CONFIG.getApiUrl("moviesPopular")
            };
            $http(req).success(function (response) {
                service.data.movies = response.results;
            }).error(function (data, status, headers, config) {
                console.error("error");
            });
        };
        service.getMovieById = function (id) {
            //TODO
        };
        service.getMovies();
    })();
});
```

Error callback

Defines what happens if we dont get data

MOVIE SERVICE

www/model/movies-service.js

```
app.factory('MoviesService', function ($http, APP_CONFIG) {
    return new (function () {
        var service = this;
        service.data = {};
        service.getMovies = function () {
            var req = {
                method: "GET",
                url: APP_CONFIG.getApiUrl("moviesPopular")
            };
            $http(req).success(function (response) {
                service.data.movies = response.results;
            }).error(function (data, status, headers, config) {
                console.error("error");
            });
        };
        service.getMovieById = function (id) {
            //TODO
        };
        service.getMovies();
    })();
});
```

Call function

GETTING MOVIE BY ID

www/model/movies-service.js

```
service.getMovieById = function (id) {
    var selectedMovie = {};
    angular.forEach(service.data.movies, function (movie) {
        if (movie.id == id) selectedMovie = movie;
    });
    return selectedMovie;
};
```

GETTING MOVIE BY ID

www/model/movies-service.js

```
service.getMovieById = function (id) {
    var selectedMovie = {};
    angular.forEach(service.data.movies, function (movie) {
        if (movie.id == id) selectedMovie = movie;
    });
    return selectedMovie;
};
```

Simple foreach for searching by movie id

MOVIE DETAIL

DEFINING APP STATES

www/app/routes.js

```
app.config(function ($stateProvider, $urlRouterProvider) {
    $stateProvider
        .state('movies', {
            url: '/movies',
            templateUrl: "views/movies/movies-list.html",
            controller: 'MoviesListCtrl'
        })
        //.state('movies-top-rated', {
        //    url: '/movies/top-rated',
        //    templateUrl: "views/movies/movies-list.html",
        //    controller: 'MoviesTopRatedCtrl'
        //})
        .state('movies-detail', {
            url: '/movies/:id',
            templateUrl: "views/movies/movies-detail.html",
            controller: 'MoviesDetailCtrl'
        });

    $urlRouterProvider.otherwise('/movies');
});
```

PREPARE

- **Create files**
  - www/views/movies/movies-detail.html
  - www/views/movies/movies-detail.js
- **Add to index.html**

```
<script src="views/movies/movies-detail.js"></script>
```

CREATING MOVIE DETAIL

www/views/movies/movies-detail.js

```
app.controller('MoviesDetailCtrl', function ($scope, MoviesService,
    $stateParams) {
    if($stateParams){
        var movieId = +$stateParams.id;
        $scope.movie = MoviesService.getMovieById(movieId);
    }
})
```

CREATING MOVIE DETAIL

www/views/movies/movies-detail.js

```
app.controller('MoviesDetailCtrl', function ($scope, MoviesService,
    $stateParams) {
    if($stateParams){
        var movieId = +$stateParams.id;
        $scope.movie = MoviesService.getMovieById(movieId);
    }
})
```

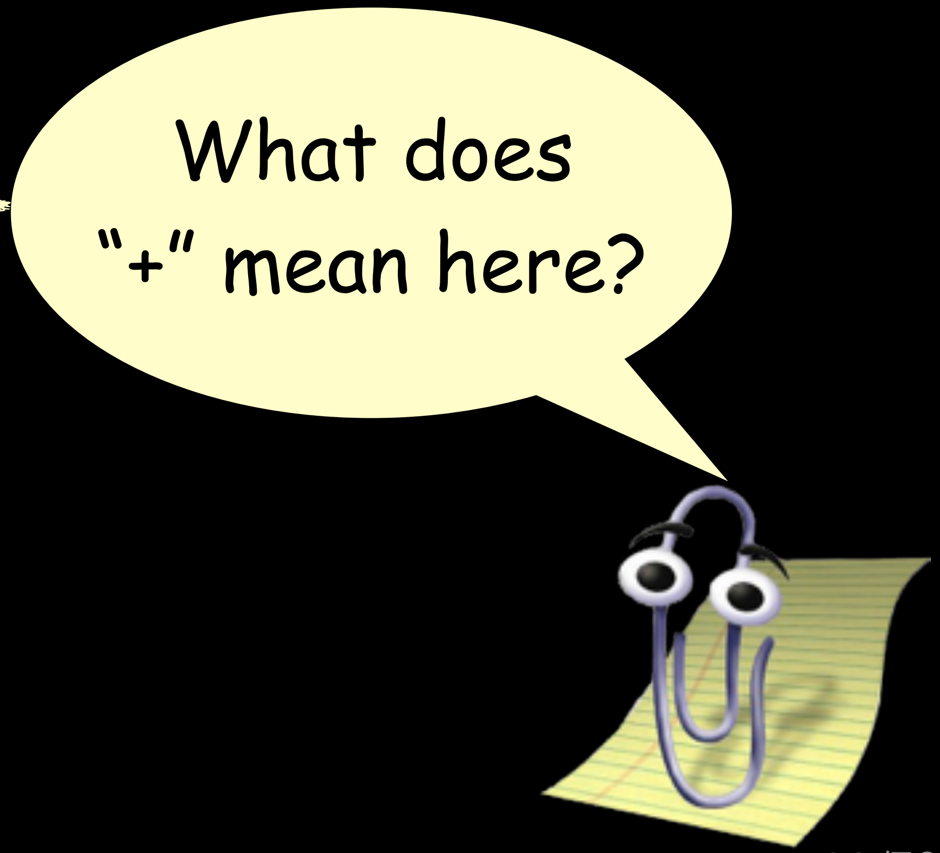
$stateParams gives URL params

Example: $stateParams.id: /movies/123 gives "123"

CREATING MOVIE DETAIL

www/views/movies/movies-detail.js

```
app.controller('MoviesDetailCtrl', function ($scope, MoviesService,
    $stateParams) {
    if($stateParams){
        var movieId = +$stateParams.id;
        $scope.movie = MoviesService.getMovieById(movieId);
    }
})
```



CREATING MOVIE DETAIL

www/views/movies/movies-detail.html

```
<ion-view title="{{movie.title}}" id="view-movies-detail">
    <ion-content>
        <img src=“http://image.tmdb.org/t/p/w500{{movie.backdrop_path}}" />
        <ul class="list">
            <li class="item">
                <strong>Release date:</strong>
                {{movie.release_date | date: 'yyyy'}}
            </li>
            <li class="item">
                <strong>Rating:</strong>
                {{movie.vote_average}} ({{movie.vote_count}}x)
            </li>
            <li class="item item-body">
                <p>{{movie.overview}}</p>
            </li>
        </ul>
    </ion-content>
</ion-view>
```

CREATING MOVIE DETAIL

www/views/movies/movies-detail.html

```
<ion-view title="{{movie.title}}" id="view-movies-detail">
    <ion-content>
        <img src=“http://image.tmdb.org/t/p/w500{{movie.backdrop_path}}" />
        <ul class="list">
            <li class="item">
                <strong>Release date:</strong>
                {{movie.release_date | date: 'yyyy'}}
            </li>
            <li class="item">
                <strong>Rating:</strong>
                {{movie.vote_average}} ({{movie.vote_count}}x)
            </li>
            <li class="item item-body">
                <p>{{movie.overview}}</p>
            </li>
        </ul>
    </ion-content>
</ion-view>
```

We know this URL from API docs

CREATING MOVIE DETAIL

www/views/movies/movies-detail.html

```
<ion-view title="{{movie.title}}" id="view-movies-detail">
    <ion-content>
        <img src=“http://image.tmdb.org/t/p/w500{{movie.backdrop_path}}" />
        <ul class="list">
            <li class="item">
                <strong>Release date:</strong>
                {{movie.release_date | date: 'yyyy'}}
            </li>
            <li class="item">
                <strong>Rating:</strong>
                {{movie.vote_average}} ({{movie.vote_count}}x)
            </li>
            <li class="item item-body">
                <p>{{movie.overview}}</p>
            </li>
        </ul>
    </ion-content>
</ion-view>
```

Filter for getting year

Example: 2016 from format 2016-01-28

UPDATE MOVIE LIST

www/views/movies/movies-list.html

```
<ion-view title="Movie list" id="view-movies-list">
    <ion-content>
        <div class="bar item-input-inset bar-search">
            <label class="item-input-wrapper">
                <i class="icon ion-ios-search"></i>
                <input type="search" ng-model="searchString" placeholder="Search">
            </label>
        </div>
        <ion-list>
            <ion-item ng-repeat="movie in data.movies | orderBy: '-vote_average' |
                    filter: searchString” class=“item-avatar"
                    ui-sref="movies-detail({'id': movie.id})">
                <img src="http://image.tmdb.org/t/p/w500{{movie.backdrop_path}}"
                    class="fit-thumbnail"/>
                <h2>{{movie.title}}</h2>
                <p>{{movie.vote_average}} / 10</p>
            </ion-item>
        </ion-list>
    </ion-content>
</ion-view>
```

UPDATE MOVIE LIST

www/views/movies/movies-list.html

```
<ion-view title="Movie list" id="view-movies-list">
    <ion-content>
        <div class="bar item-input-inset bar-search">
            <label class="item-input-wrapper">
                <i class="icon ion-ios-search"></i>
                <input type="search" ng-model="searchString" placeholder="Search">
            </label>
        </div>
        <ion-list>
            <ion-item ng-repeat="movie in data.movies | orderBy: '-vote_average' |
                    filter: searchString” class=“item-avatar"
                    ui-sref="movies-detail({'id': movie.id})">
                <img src="http://image.tmdb.org/t/p/w500{{movie.backdrop_path}}"
                    class="fit-thumbnail"/>
                <h2>{{movie.title}}</h2>
                <p>{{movie.vote_average}} / 10</p>
            </ion-item>
        </ion-list>
    </ion-content>
</ion-view>
```

Link to Movie detail
passing movie id

UPDATE MOVIE LIST

www/views/movies/movies-list.html

```
<ion-view title="Movie list" id="view-movies-list">
    <ion-content>
        <div class="bar item-input-inset bar-search">
            <label class="item-input-wrapper">
                <i class="icon ion-ios-search"></i>
                <input type="search" ng-model="searchString" placeholder="Search">
            </label>
        </div>
        <ion-list>
            <ion-item ng-repeat="movie in data.movies | orderBy: '-vote_average' |
                    filter: searchString” class=“item-avatar"
                    ui-sref="movies-detail({'id': movie.id})">
                <img src="http://image.tmdb.org/t/p/w500{{movie.backdrop_path}}"
                    class="fit-thumbnail"/>
                <h2>{{movie.title}}</h2>
                <p>{{movie.vote_average}} / 10</p>
            </ion-item>
        </ion-list>
    </ion-content>
</ion-view>
```

Order by
movie.vote_average descending

UPDATE MOVIE LIST

Avatar needs class item-avatar

www/views/movies/movies-list.html

```
<ion-view title="Movie list" id="view-movies-list">
    <ion-content>
        <div class="bar item-input-inset bar-search">
            <label class="item-input-wrapper">
                <i class="icon ion-ios-search"></i>
                <input type="search" ng-model="searchString" placeholder="Search">
            </label>
        </div>
        <ion-list>
            <ion-item ng-repeat="movie in data.movies | orderBy: '-vote_average' |
                    filter: searchString” class=“item-avatar"
                    ui-sref="movies-detail({'id': movie.id})">
                <img src="http://image.tmdb.org/t/p/w500{{movie.backdrop_path}}"
                    class="fit-thumbnail"/>
                <h2>{{movie.title}}</h2>
                <p>{{movie.vote_average}} / 10</p>
            </ion-item>
        </ion-list>
    </ion-content>
</ion-view>
```

Add avatar

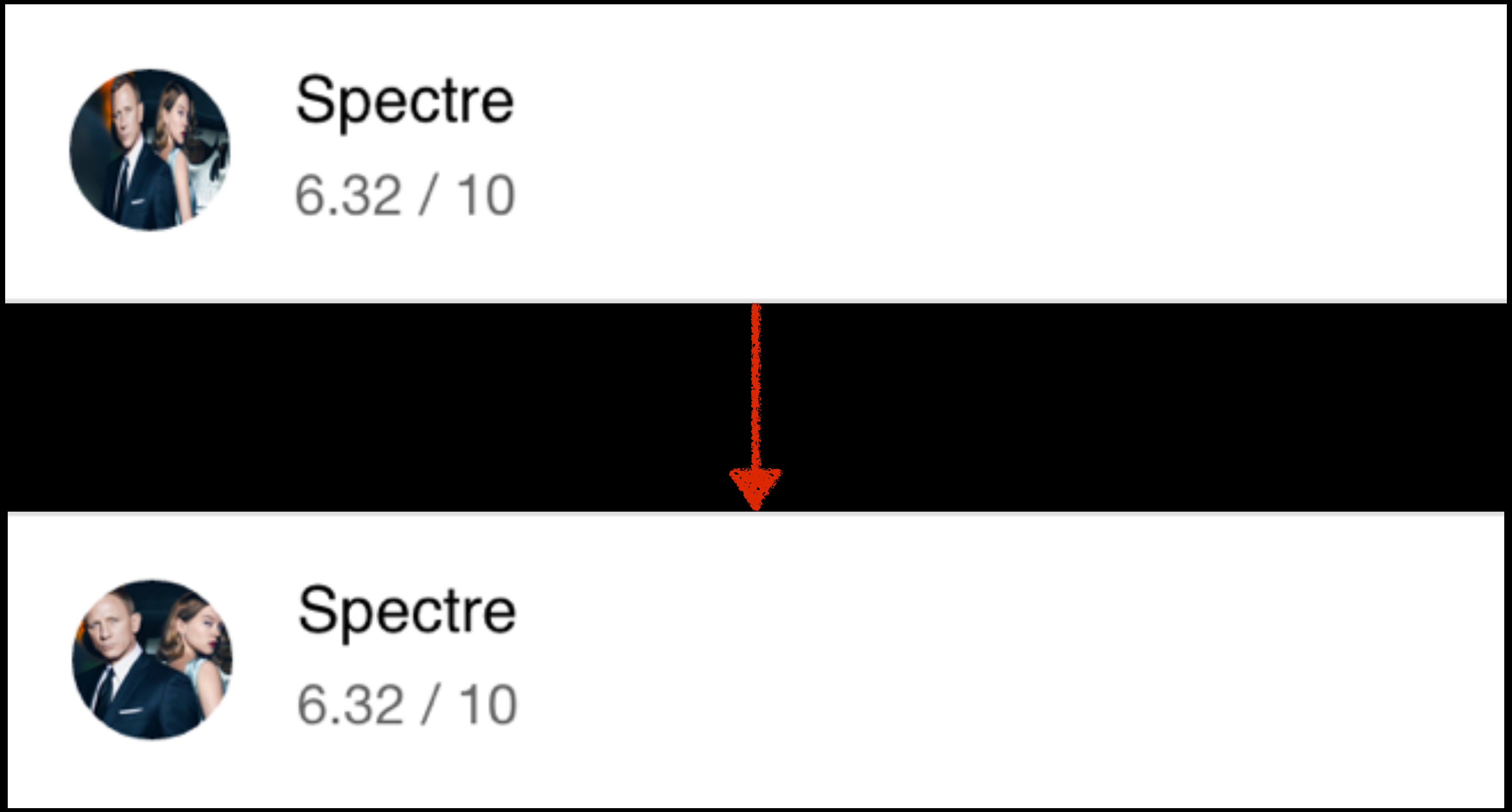
UPDATE MOVIE LIST

www/views/movies/movies-list.html

```
<ion-view title="Movie list" id="view-movies-list">
    <ion-content>
        <div class="bar item-input-inset bar-search">
            <label class="item-input-wrapper">
                <i class="icon ion-ios-search"></i>
                <input type="search" ng-model="searchString" placeholder="Search">
            </label>
        </div>
        <ion-list>
            <ion-item ng-repeat="movie in data.movies | orderBy: '-vote_average' |
                    filter: searchString” class=“item-avatar"
                    ui-sref="movies-detail({'id': movie.id})">
                <img src="http://image.tmdb.org/t/p/w500{{movie.backdrop_path}}"
                    class="fit-thumbnail"/>
                <h2>{{movie.title}}</h2>
                <p>{{movie.vote_average}} / 10</p>
            </ion-item>
        </ion-list>
    </ion-content>
</ion-view>
```

Add vote average

SCSS IMAGE RATIO FIX



Spectre

6.32 / 10

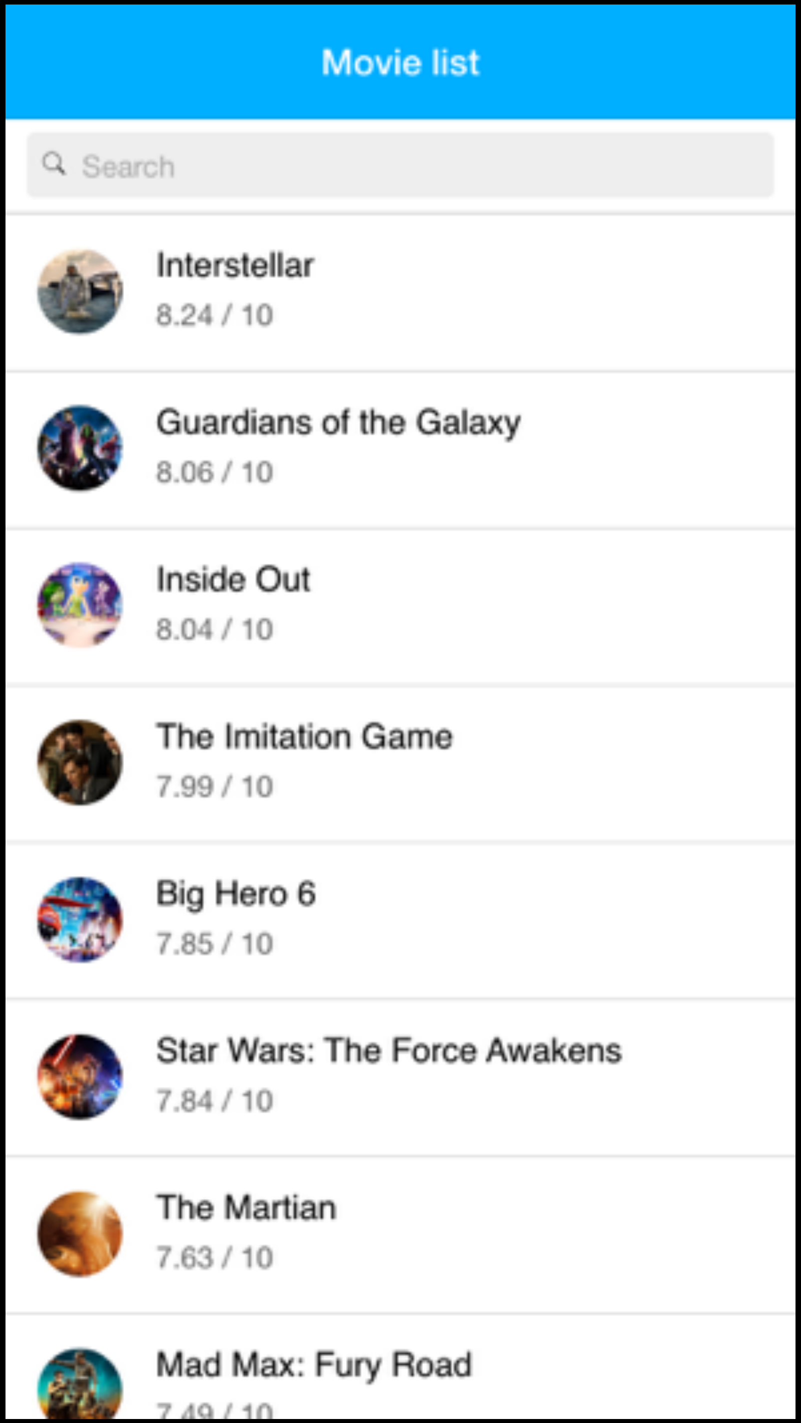
Spectre

6.32 / 10

scss/views/_common.scss

```scss
.fit-thumbnail {
  object-fit: cover;
}
```

YOU SHOULD SEE THIS



Movie list

Search

Interstellar
8.24 / 10

Guardians of the Galaxy
8.06 / 10

Inside Out
8.04 / 10

The Imitation Game
7.99 / 10

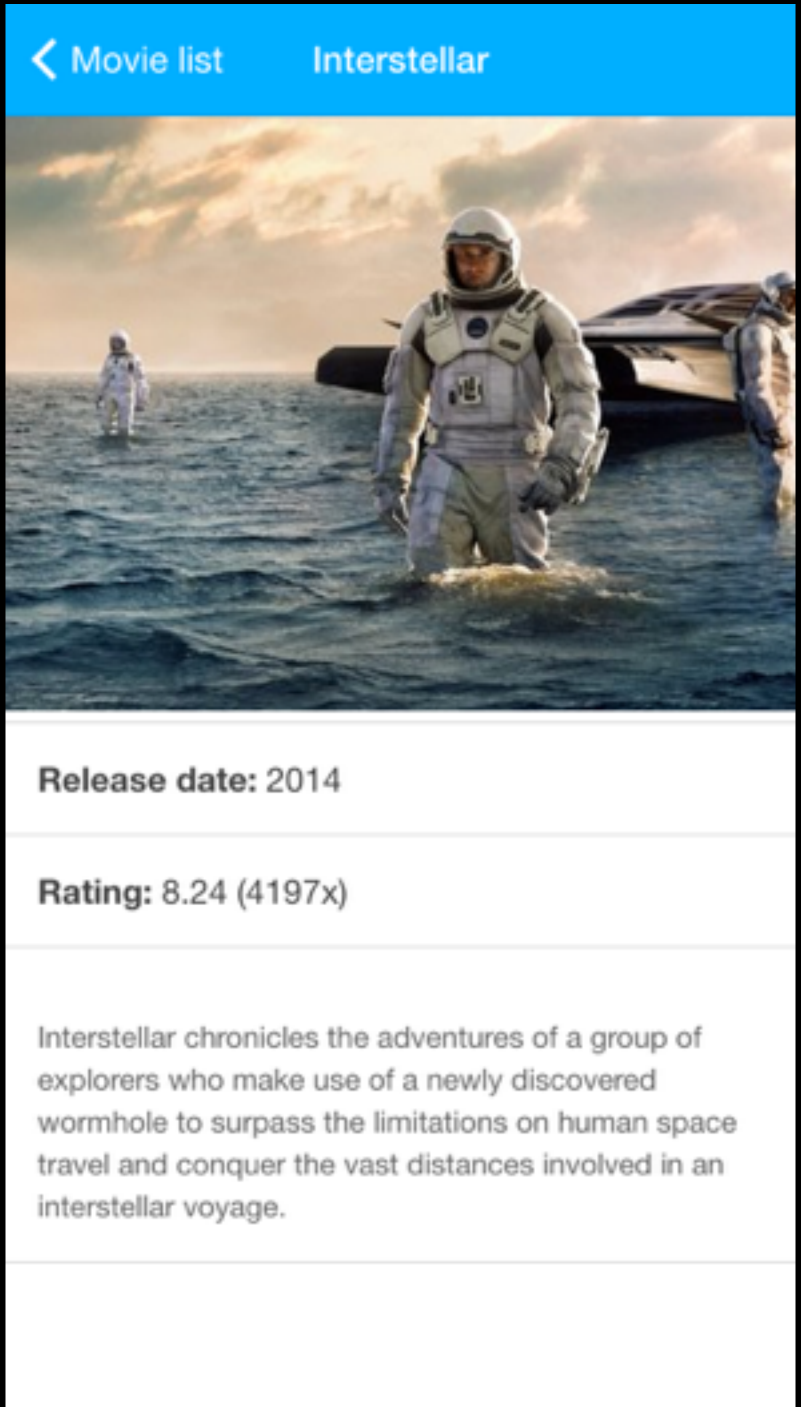
Big Hero 6
7.85 / 10

Star Wars: The Force Awakens
7.84 / 10

The Martian
7.63 / 10

Mad Max: Fury Road
7.49 / 10

AND IF WE CLICK ON "INTERSTELLAR"



Release date: 2014

Rating: 8.24 (4197x)

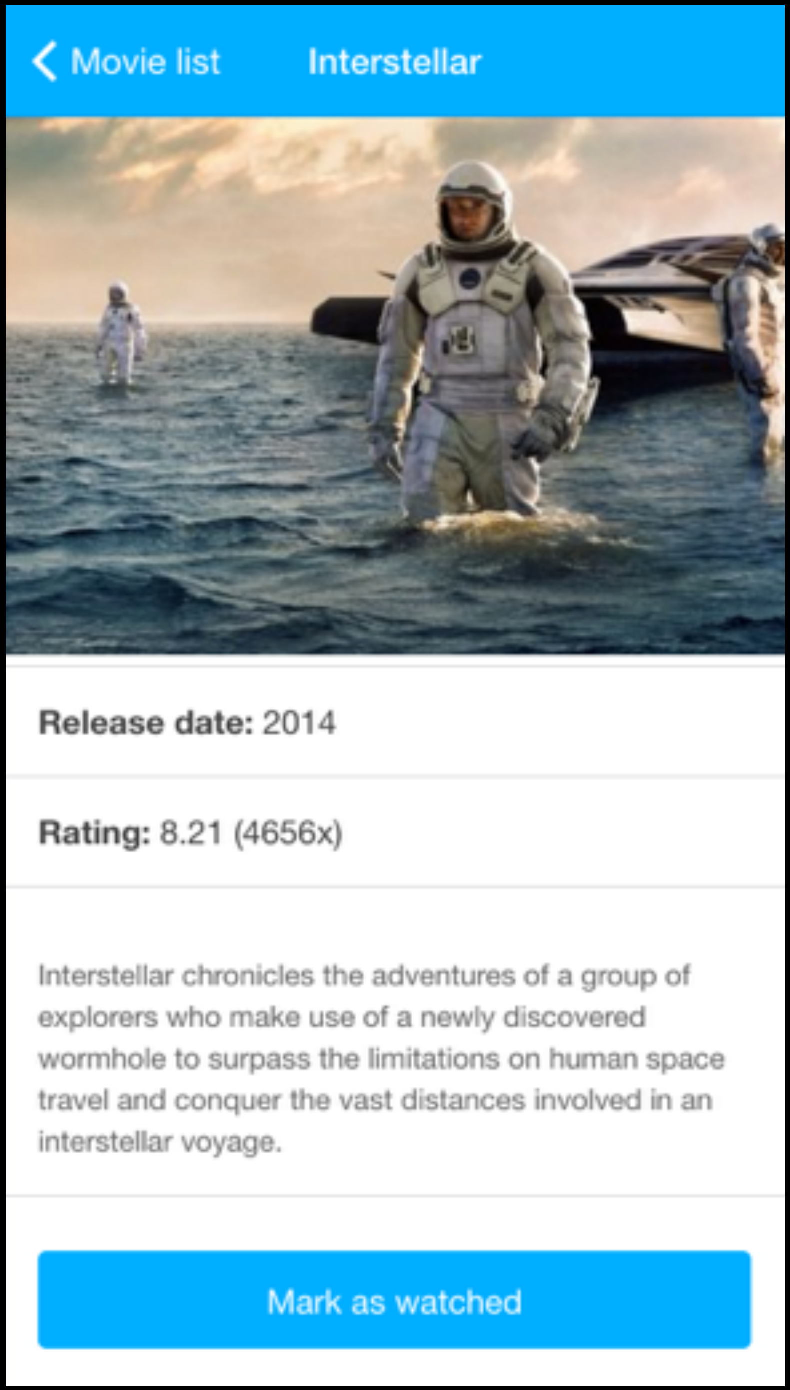
Interstellar chronicles the adventures of a group of explorers who make use of a newly discovered wormhole to surpass the limitations on human space travel and conquer the vast distances involved in an interstellar voyage.

MARK MOVIE AS WATCHED

MARK MOVIE AS WATCHED

- We will use Local Storage (not the best choice)
- Add button in Movie detail

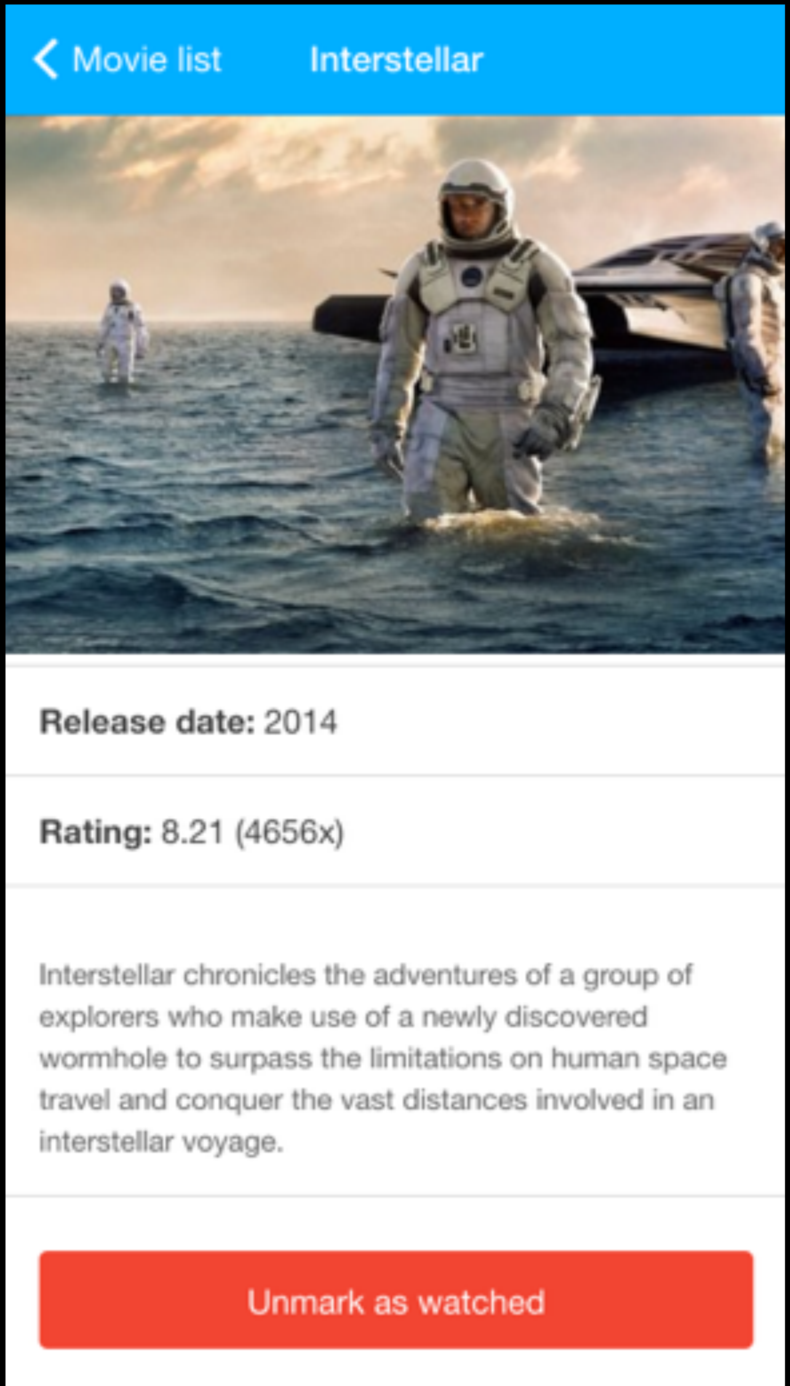
EXPECTED RESULT



Release date: 2014

Rating: 8.21 (4656x)

Interstellar chronicles the adventures of a group of explorers who make use of a newly discovered wormhole to surpass the limitations on human space travel and conquer the vast distances involved in an interstellar voyage.



Release date: 2014

Rating: 8.21 (4656x)

Interstellar chronicles the adventures of a group of explorers who make use of a newly discovered wormhole to surpass the limitations on human space travel and conquer the vast distances involved in an interstellar voyage.

GET WATCHED MOVIES AFTER LOAD MOVIES

www/model/movies-service.js

```
service.getMovies = function (callback) {
    var req = {
        method: "GET",
        url: APP_CONFIG.getApiUrl("moviesPopular")
    };
    $http(req).success(function (response) {
        service.data.movies = response.results;
        if (callback) callback();
    }).error(function (data, status, headers, config) {
        console.error("error");
    });
};

service.getMovies(function(){
    service.getWatchedMoviesFromStorage();
});
```

GET WATCHED MOVIES AFTER LOAD MOVIES

www/model/movies-service.js

```
service.getMovies = function (callback) {
    var req = {
        method: "GET",
        url: APP_CONFIG.getApiUrl("moviesPopular")
    };
    $http(req).success(function (response) {
        service.data.movies = response.results;
        if (callback) callback();
    }).error(function (data, status, headers, config) {
        console.error("error");
    });
};

service.getMovies(function(){
    service.getWatchedMoviesFromStorage();
});
```

Call function in callback

GET DATA FROM LOCAL STORAGE

www/model/movies-service.js

```
service.watchedMovies = [];

service.getWatchedMoviesFromStorage = function () {
    try {
        service.watchedMovies = JSON.parse(
          localStorage.getItem("watched_movies")
        ) || [];
    } catch (e) {
        console.warn("Invalid JSON string")
    }
    return service.watchedMovies;
};
```

GET DATA FROM LOCAL STORAGE

www/model/movies-service.js

```
service.watchedMovies = [];

service.getWatchedMoviesFromStorage = function () {
    try {
        service.watchedMovies = JSON.parse(
          localStorage.getItem("watched_movies")
        ) || [];
    } catch (e) {
        console.warn("Invalid JSON string")
    }
    return service.watchedMovies;
};
```

GET DATA FROM LOCAL STORAGE

www/model/movies-service.js

```
service.watchedMovies = [];

service.getWatchedMoviesFromStorage = function () {
    try {
        service.watchedMovies = JSON.parse(
          localStorage.getItem("watched_movies")
        ) || [];
    } catch (e) {
        console.warn("Invalid JSON string")
    }
    return service.watchedMovies;
};
```

"Not everything is a JSON document"
- Old Chinese proverb

GET DATA FROM LOCAL STORAGE

www/model/movies-service.js

```
service.watchedMovies = [];

service.getWatchedMoviesFromStorage = function () {
    try {
        service.watchedMovies = JSON.parse(
          localStorage.getItem("watched_movies")
        ) || [];
    } catch (e) {
        console.warn("Invalid JSON string")
    }
    return service.watchedMovies;
};
```

Happens when JSON is not valid

GET DATA FROM LOCAL STORAGE

www/model/movies-service.js

```
service.watchedMovies = [];

service.getWatchedMoviesFromStorage = function () {
    try {
        service.watchedMovies = JSON.parse(
          localStorage.getItem("watched_movies")
        ) || [];
    } catch (e) {
        console.warn("Invalid JSON string")
    }
    return service.watchedMovies;
};
```

When JSON is empty returns empty array

TOGGLE BUTTON FOR WATCHED MOVIES

www/model/movies-service.js

```
service.toggleWatched = function (id) {
    var occurenceIndex = service.watchedMovies.indexOf(id);
    if (~occurenceIndex)
        service.watchedMovies.splice(occurenceIndex, 1);
    else
        service.watchedMovies.push(+id);
    localStorage.setItem("watched_movies", JSON.stringify(
        service.watchedMovies)
    );
};
```

TOGGLE BUTTON FOR WATCHED MOVIES

www/model/movies-service.js

```
service.toggleWatched = function (id) {
    var occurenceIndex = service.watchedMovies.indexOf(id);
    if (~occurenceIndex)
        service.watchedMovies.splice(occurenceIndex, 1);
    else
        service.watchedMovies.push(+id);
    localStorage.setItem("watched_movies", JSON.stringify(
        service.watchedMovies)
    );
};
```

If there is an occurrence
remove item from storage and reverse

CHECK IF MOVIE IS WATCHED

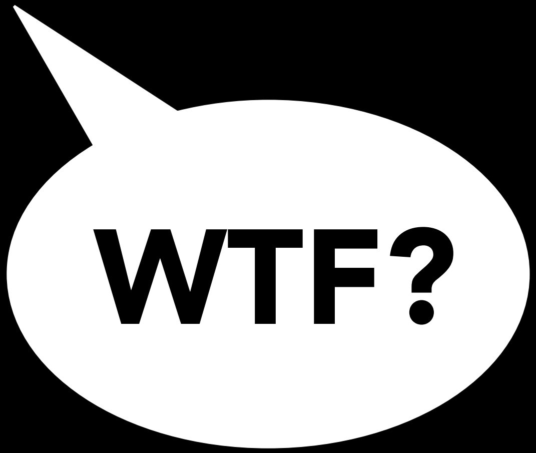
`www/model/movies-service.js`

```
service.isMovieWatched = function (movieId) {
    return !!~service.watchedMovies.indexOf(movieId)
};
```

CHECK IF MOVIE IS WATCHED

www/model/movies-service.js

```
service.isMovieWatched = function (movieId) {
    return !!~service.watchedMovies.indexOf(movieId)
};
```



CHECK IF MOVIE IS WATCHED

www/model/movies-service.js

```
service.isMovieWatched = function (movieId) {
    return !!~service.watchedMovies.indexOf(movieId)
};
```

Returns position of "id" in Array
or "-1" if not found

CHECK IF MOVIE IS WATCHED

www/model/movies-service.js

```
service.isMovieWatched = function (movieId) {
    return !!~service.watchedMovies.indexOf(movieId)
};
```

"!!~" ensures that function returns
"true" if there is an occurrence or "false" if not

ADD BUTTON TO MOVIE DETAIL

www/views/movies/movies-detail.html

```
<ion-view title="{{movie.title}}" id="view-movies-detail">
    <ion-content>
        <img src="http://image.tmdb.org/t/p/w500/{{movie.backdrop_path}}" />
        <ul class="list">
            ...
            <li class="item">
                <button class="button button-block"
                    ng-click="toggleWatched(movie.id)"
                    ng-class="{
                        'button-positive': !isMovieWatched(),
                        'button-assertive': isMovieWatched()
                    }">
                    <span ng-if="!isMovieWatched()">Mark as watched</span>
                    <span ng-if="isMovieWatched()">Unmark as watched</span>
                </button>
            </li>
        </ul>
    </ion-content>
</ion-view>
```

ADD BUTTON TO MOVIE DETAIL

www/views/movies/movies-detail.html

```
<ion-view title="{{movie.title}}" id="view-movies-detail">
    <ion-content>
        <img src="http://image.tmdb.org/t/p/w500/{{movie.backdrop_path}}" />
        <ul class="list">
            ...
            <li class="item">
                <button class="button button-block"
                    ng-click="toggleWatched(movie.id)"
                    ng-class="{
                        'button-positive': !isMovieWatched(),
                        'button-assertive': isMovieWatched()
                    }">
                    <span ng-if="!isMovieWatched()">Mark as watched</span>
                    <span ng-if="isMovieWatched()">Unmark as watched</span>
                </button>
            </li>
        </ul>
    </ion-content>
</ion-view>
```

Call markAsWatched(movie.id) after click on button

ADD BUTTON TO MOVIE DETAIL

www/views/movies/movies-detail.html

```
<ion-view title="{{movie.title}}" id="view-movies-detail">
    <ion-content>
        <img src="http://image.tmdb.org/t/p/w500/{{movie.backdrop_path}}" />
        <ul class="list">
            ...
            <li class="item">
                <button class="button button-block"
                    ng-click="toggleWatched(movie.id)"
                    ng-class="{
                        'button-positive': !isMovieWatched(),
                        'button-assertive': isMovieWatched()
                    }">
                    <span ng-if="!isMovieWatched()">Mark as watched</span>
                    <span ng-if="isMovieWatched()">Unmark as watched</span>
                </button>
            </li>
        </ul>
    </ion-content>
</ion-view>
```

Add class "button-positive" when !movieWatched and reverse

BIND CLICK ON BUTTON

www/views/movies/movies-detail.js

```
app.controller('MoviesDetailCtrl', function ($scope, MoviesService,
    $stateParams) {

    if($stateParams){
        var movieId = +$stateParams.id;
        $scope.movie = MoviesService.getMovieById(movieId);
    }

    $scope.toggleWatched = function(){
        MoviesService.toggleWatched(movieId);
    };

    $scope.isMovieWatched = function () {
        return MoviesService.isMovieWatched(movieId);
    };
})
```

BIND CLICK ON BUTTON

www/views/movies/movies-detail.js

```
app.controller('MoviesDetailCtrl', function ($scope, MoviesService,
    $stateParams) {

    if($stateParams){
        var movieId = +$stateParams.id;
        $scope.movie = MoviesService.getMovieById(movieId);
    }

    $scope.toggleWatched = function(){
        MoviesService.toggleWatched(movieId);
    };

    $scope.isMovieWatched = function () {
        return MoviesService.isMovieWatched(movieId);
    };
})
```

Bind click on button

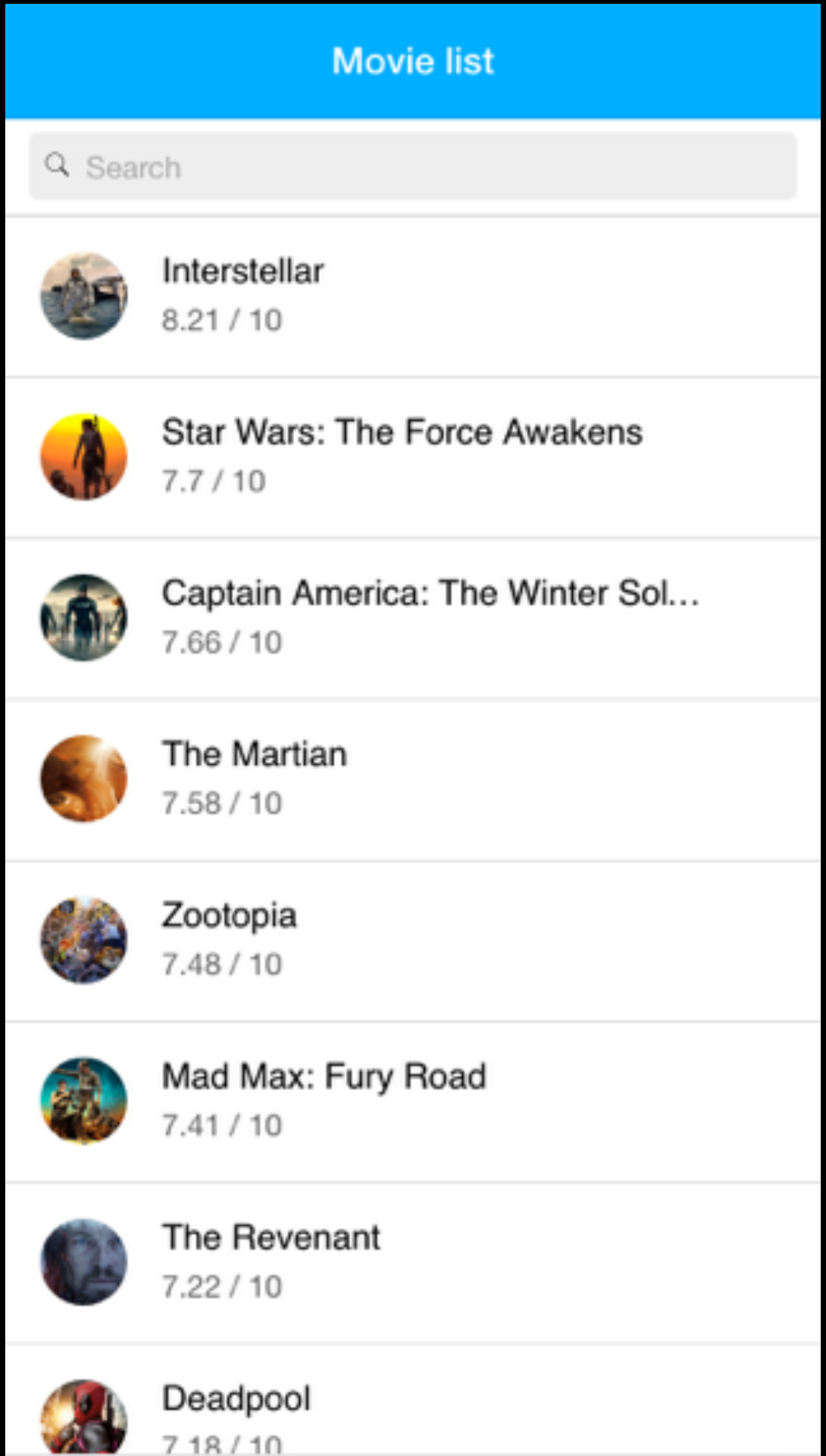
BIND CLICK ON BUTTON

www/views/movies/movies-detail.js

```
app.controller('MoviesDetailCtrl', function ($scope, MoviesService,
    $stateParams) {

    if($stateParams){
        var movieId = +$stateParams.id;
        $scope.movie = MoviesService.getMovieById(movieId);
    }

    $scope.toggleWatched = function(){
        MoviesService.toggleWatched(movieId);
    };

    $scope.isMovieWatched = function () {
        return MoviesService.isMovieWatched(movieId);
    };
})
```

Check if movie with movieId is watched

MOVIE LIST



Movie list

Search

Interstellar
8.21 / 10

Star Wars: The Force Awakens
7.7 / 10

Captain America: The Winter Sol...
7.66 / 10

The Martian
7.58 / 10

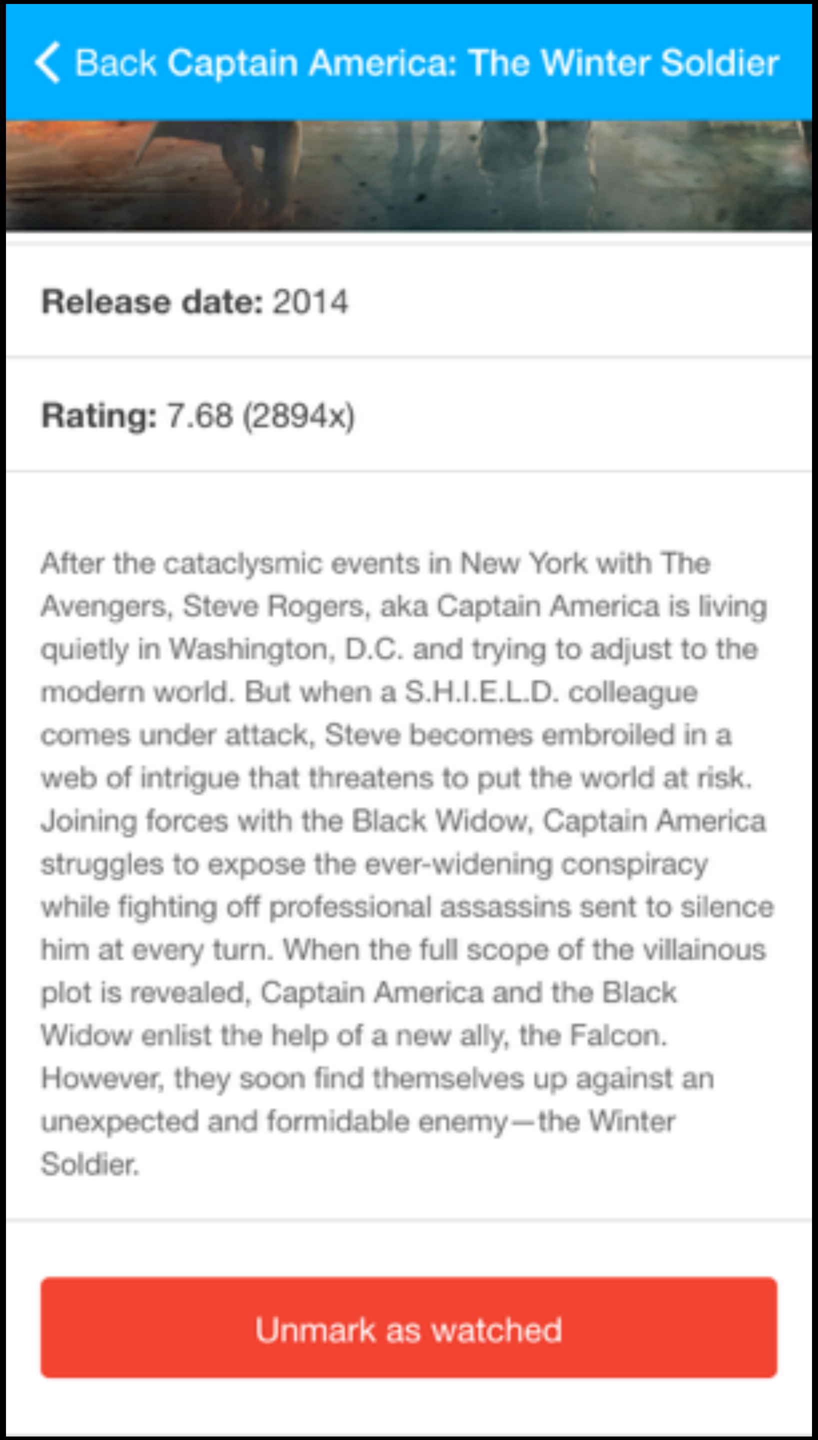
Zootopia
7.48 / 10

Mad Max: Fury Road
7.41 / 10

The Revenant
7.22 / 10

Deadpool
7.18 / 10

AFTER CLICK ON CAPTAIN AMERICA



Release date: 2014

Rating: 7.68 (2894x)

After the cataclysmic events in New York with The Avengers, Steve Rogers, aka Captain America is living quietly in Washington, D.C. and trying to adjust to the modern world. But when a S.H.I.E.L.D. colleague comes under attack, Steve becomes embroiled in a web of intrigue that threatens to put the world at risk. Joining forces with the Black Widow, Captain America struggles to expose the ever-widening conspiracy while fighting off professional assassins sent to silence him at every turn. When the full scope of the villainous plot is revealed, Captain America and the Black Widow enlist the help of a new ally, the Falcon. However, they soon find themselves up against an unexpected and formidable enemy—the Winter Soldier.

QUESTIONS?

HOMEWORK

Infinite scroll

http://bit.ly/1PDqUDU

http://ionicframework.com/docs/api/directive/ionInfiniteScroll/

U+_

JAN VÁCLAVÍK

@janvaclavik

DANIEL RYS

@danielrys

WWW.USERTECHNOLOGIES.COM