비트피알 ReadMe

개발환경

- 프레임웍 : AngularJs(MeanJS), ExpressJs를 사용하여 개발됨
- 언어 : javascript, css, html5
- 소스 저장소 : git - https://github.com/leorez/bitpr
- grunt cli 사용
- 테스트 프레임웍 : 클라이언트 - karma, 서버 - mocha, e2e - protractor
- 서버: 아마존 AWS linux

소스받기

github : https://github.com/leorez/bitpr

- 추천 [Fork]버튼을 눌러 fork함 - github에 로긴후 fork를 통해 자신의 계정으로 저장소를 복사합니다.
- [Clone or download] 버튼을 눌러 clone또는 다운로드함
  clone으로 기존 저장소를 사용하여 개발할경우 아이디를 알려주시면 저장소 멤버로 등록해드립니다.

소스를 받고난뒤 해야할일

```
$ npm install
$ bower install
```

디버그모드 3000번 포트로 테스트서버를 실행하기

```
$ grunt debug
```

브라우저에서 localhost:3000으로 접속한다.

테스트모드로 3001번 포트 서버실행하기

```
$ grunt testserver
```

브라우저에서 localhost:3001으로 접속한다.

실행모드의 이해

MeanJs프레임웍은 3가지 모드로실행이 가능하다. 또한 각 모드별로 독립적인DB를 사용한다.

- test - 테스트실행시 사용하는 모드 테스트실행전 모든 데이타를 삭제한후 실행된다. (db : bitpr-test)
- develop - 개발모드로써 데이타를 삭제하지않고 보존한다. 실디비와 구분되어 서비스에 영향을 주지않고 개발이 가능하다.(db : bitpr-dev)
- production - 실서비스 모드 (db : bitpr)

폴더구성

modules

뷰와 컨트롤러및 서비스등 웹을 구성하는 프로그램 모듈들이 있는곳이다.

- modulses/article-senders - 보도자료
- modules/core - 코어모듈, 전역적으로 사용되는 모듈
- modules/dashboard - 대쉬보드
- modules/mailinglist - 메일링리스트 수집및 관리(담당: 김민지)
- modules/monitoring - 기사수집목록및 집계차트
- modules/reporters - 기자관리
- modules/users - 사용자

public

웹루트 경로에 해당되는 폴더이며 static한 자료들이 저장되어있다.

- embed.js - 홈페이지 게시용 libary
- embed-test.js - 홈페이지에 올린 보도자료 리스트보기 샘플
- embed-test-search.js - 홈페이지에 올린 검색기사 리스트보기 샘플

lib

외부서비스들을 사용하기위한 라이브러리 모듈을 저장한다.

- aritle.counter.js - 기사 집계 모듈
- cafe24.smms.js - 카페24 sms 모듈
- dart.js - Dart API
- mail.js - Mailgun으로 메일보내기 모듈

config

각종 설정파일들

- assets - css, javascript lib 들을 설정
- env - 환경설정, kakao등 소셜미디어 연동관련 설정들
- lib - 서버에서 사용하는 라이브러리

tests

lib또는 scheduler.js와 같은 백그라운드 데몬등의 테스트

uploads

파일을 업로드하면 저장되는 곳

주요파일

scheduler.js

백그라운드 데몬, 5분마다 실행되며 스케쥴설정은 크론과 동일하다(크론 스케쥴설정법참조). 보도자료발송, 기사수집, 수집된 기사 집계를 처리한다.

```
// 5분마다 실행됨
var job = schedule.scheduleJob('*/5 * * * *', function () {
  exports.run();
});
```

서버에서 실행,중지하는법

```
$ pwd
$ /home/ec2-user/www
$ forever stop scheduler.js
$ forever start scheduler.js
```

public/embed.js

홈페이지에 게시하기관련모듈 public/embed-test.html(보도자료), public/embed-test-search.html(검색된기사) 열면 업체홈페이지에 embed할 sample코드가 들어있다.

```
<div class="center">------------------ Embeded start --------------------</div>

<div id="bitpr_thread"></div>
<script>
  /**
   * 기업상장코드를 변경하세요.
   * bitpr_config.corpCode
   */
  var bitpr_config = {
    corpCode: '005930',  // 기업상장코드
    type: 'REPORT', // 자료유형 : SEARCH(검색된기사), REPORT(보도자료)
    target: 'bitpr_thread', // 자료가 표시될 Element Id
    host: ''  // 수정하지 마세요.
  };

  // ----------------------------------------------
  // 아래 코드는 변경하지 마십시요.
  (function () {
    var d = document, s = d.createElement('script');
    s.src = bitpr_config.host + '/embed.js';
    s.crossorigin = 'anonymous';
    s.setAttribute('data-timestamp', +new Date());
    (d.head || d.body).appendChild(s);
  })();
  // ----------------------------------------------
</script>
<noscript>자바스크립트를 활성화 해주세요. (주)비트피알</noscript>

<div>------------------ Embeded end --------------------</div>
```

> embed.js 테스트 방법 debug모드로 서버를 실행후 브라우져에서
> http://localhost:3000/embed-test.html 실행

- protractor.conf.js - e2e 테스트 설정파일
- karma.conf.js - 클라이어트 테스트 설정파일
- gruntfile.js - grunt 설정파일

서버

- AWS Linux
- 서버접속용 pem파일 - birpr.pem (보안상 메일로 전달)

서버접속

1. bitpr.pem파일을 소스 루트로옮긴다.
2. 소스루트에서 다음과 같이 실행

```
$ server.sh
```

배포

로컬에서 master로 git push한후 서버에서 다음과 같이 실행한다.

```
$ git pull origin master
$ ./deploy.sh
```

개발절차

디버그모드로 서버실행

```
$ grunt debug
```

http://localhost:3000/ 접속

이렇게하면 file watcher가 같이 실행되어 파일을 수정시 자동으로 화면이 갱신되어 변경사항을 바로 확인할수 있다. 또한 테스트파일을 수정하면 테스트가 실행된다.

테스트

E2E 테스트

브라우져를 이용한 자동테스트

> 주의) 로그인이 필요한 테스트와 필요하지 않은 테스트로 구분된다. user테스트는 로그인이 필요하지 않다.

protractor.conf.js에 다음을 true로하면 로그인을 먼저 실행한후 테스트를 진행한다.
user테스트의(grunt test:e2e -suite users) 경우에는 로그인을 먼저하지 않으므로 false로 한다.

```
var needSignin = false;
```

e2e테스트 실행

```
$ grunt test:e2e -suite suite명

example
$ grunt test:e2e -suite senders
$ grunt test:e2e -suite users

suite는 protractor.conf.js참조
```

테스트파일경로
/config/assets/test.js참조

클라이언트 테스트

```
grunt test:client
```

서버 테스트

```
grunt test:server
```

모든 테스트

```
grunt test:server
```

스케쥴러 테스트

```
$ grunt test:runScheduler

file watch mode
새로운 콘솔창에서
$ grunt debug
or
$ grunt watch

직접실행
새로운 콘솔창에서
$ grunt test:scheduler
```

실행하면 tests/scheduler-test.js가 실행되면서 테스트 데이타를 bitpr-test 데이타베이스에 저장하게되고
grunt test:runScheduler 으로 실행되고 있던 스케쥴러가 데이타를 발견하면 그에맞게 자동실행된다.
file watch mode 에서는 scheduler.js나 tests/scheduler-test.js가 수정되면 자동 테스트가 진행된다.

> dev:runScheduler development모드에서 테스트할경우 같은 모드로 스케쥴러를 실행한다.

기사집계 테스트

```
$ grunt watch

새로운 콘솔창에서
$ grunt test:articleCounter
$ grunt testserver
```

- 관련파일 - /lib/article.counter.js, /tests/article.counter.spec.js, scheduler.js
- test모드로 실행되므로 testserver를 실행해서 http://localhost:3001/ 로 접속하면 테스트 데이타를 확인할수 있다.

cafe24 sms 테스트

```
$ grunt test:sms
```

- 관련파일 - /lib/cafe24.sms.js /tests/cafe24.sms.spec.js

mail 테스트

```
$ grunt test:mail
```

- 관련파일 - /lib/mail.js /tests/mail.spec.js

dart 테스트

```
$ grunt test:dart
```

- 관련파일 - /lib/dart.js /tests/dart.spec.js

라이브러리 설치

클라이언트용 라이브러리 설치

```
bower install --save 라이브러리
```

설치후 /config/assets/default.js와 /config/assets/production.js파일을 수정해야한다.
/config/assets/default.js, product.js에 설치한 모듈의 css와 js경로를 추가한다.
MeanJs에서는 index.html에 직접추가하지않고 설정파일에 넣는 방식을 사용한다.

서버용라이브러리 설치

```
npm install --save 라이브러리
```

인증

상장코드인증

Dart API를 통해 상장코드를 조회하고 매칭되는 기업이 있는지 확인한다.

이메일인증

회원가입시 이메일로 인증메일을 보내 인증받는다.

전화인증

회원가입을 완료하면 회원은 guest권한이 주어지며 관리자가 전화를 걸어 확인한후 사용자관리 수정에서 전화인증을 체크하면 user:정식사용자로 변경된다. 이때부터 서비스를 사용할수 있게된다.

쇼셜미디어 연동

파일

- /modules/users/client/controllers/authentication.client.controller.js
- /modules/users/server/routes/auth.server.routes.js
- /modules/users/server/controllers/users/users.authentication.server.controller.js

라우트

/modules/users/server/routes/auth.server.routes.js

```
// Setting the facebook oauth routes
app.route('/api/auth/facebook').get(users.oauthCall('facebook', {
  scope: ['email']
}));
app.route('/api/auth/facebook/callback').get(users.oauthCallback('facebook'));

// Setting the kakao oauth routes
app.route('/api/auth/kakao').get(users.oauthCall('kakao'));
app.route('/api/auth/kakao/callback').get(users.oauthCallback('kakao'));

// Setting the naver oauth routes
app.route('/api/auth/naver').get(users.oauthCall('naver'));
app.route('/api/auth/naver/callback').get(users.oauthCallback('naver'));

// Setting the naver oauth routes
app.route('/api/auth/daum').get(users.oauthCall('daum'));
app.route('/api/auth/daum/callback').get(users.oauthCallback('daum'));
```

관련 함수

- oauthCall
- oauthCallback
- saveOAuthUserProfile

APP_ID 설정

/config/env/production.js , /config/env/test.js
현재 사용되고 있는 APP_ID는 bitpr계정용 APP_ID로 교체해야합니다.

```
daum: {
    clientID: process.env.DAUM_ID || '7435275087308742551',
    clientSecret: process.env.DAUM_SECRET || '8b1fc351505fff8c0f3b8363317c9d5d',
    callbackURL: '/api/auth/daum/callback'
  },
  naver: {
    clientID: process.env.NAVER_ID || 'G1gAlVr0vgGNOCzprgY8',
    clientSecret: process.env.NAVER_SECRET || 'd9MsJTW6ex',
    callbackURL: '/api/auth/naver/callback'
  },
  kakao: {
    clientID: process.env.KAKAO_ID || '0801e3f49b5368fab6affab03f069572',
    callbackURL: '/api/auth/kakao/callback'
  },
  facebook: {
    clientID: process.env.FACEBOOK_ID || 'APP_ID',
    clientSecret: process.env.FACEBOOK_SECRET || 'APP_SECRET',
    callbackURL: '/api/auth/facebook/callback'
  },
```

TODO

- 상장코드 인증시 중복여부확인
- 이메일 수정시 이메일 입력창띄우기 (수정화면에서 직접수정할수 없도록 ReadOnly로함)
- 이메일 수정시 중복여부확인
- 보도자료 발송 상태 자동업데이트
- [스케쥴러] 기사수집시 중복(3개씩) 저장되는 버그
- [스케쥴러] 1만명이상 유저 테스트
- [기사모니터링] 키워드별로 분류해서 보여주기
- [기자관리] 우선순위 설정하기 (드래그앤 드랍)
- [소셜미디어연동] 개발자계정 변경하기
- [소셜미디어연동] 페이스북 연동테스트 (페이스북에서 APP_ID를 받아서 테스트)

README to PDF

```
$ npm install -g markdown-pdf
$ markdown-pdf README.md
```

개발문의: noruya@gmail.com