%SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System32\Wbem;%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\;C:\Program Files\Intel\DMIX;C:\Program Files\OpenVPN\bin;C:\Program Files\Git\bin;C:\Program Files\MongoDB\Server\3.0\bin;C:\Program Files\Java\jdk1.8.0\_65\bin;D:\gradle-2.11-all\gradle-2.11\bin;C:\Program Files (x86)\Skype\Phone\;C:\Program Files\nodejs\

Testing

Mongo:

|  |  |
| --- | --- |
| Inserts | db.users.insert({"username":"Diego","password":"123","roles": [ "process\_starter" ] })  db.users.insert({"username":"Virginia","password":"123","roles": [ "process\_starter" ] })  db.users.insert({"username":"Guillermo","password":"123","roles": [ "process\_starter" ] }) |
| Updates | db.users.update({username : "Diego"}, {$set : {"password" : "a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3"  db.users.update({username : "Virginia"}, {$set : {"password" : "a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3"  db.users.update({username : "Guillermo"}, {$set : {"password" : "a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3"  db.process.update({"ObjectId":"56e96df110e0308fcec75031"},{"description":"Este trámite tiene como finalidad obtener la habilitación de un Vehículo adicional para una empresa de transportes ya inscripta en el régimen municipal. Si se trata de la inscripción de una o más unidades para el inicio de la actividad comercial (por ejemplo taxiflet), se tramita como una Inscripción/Habilitación. Si se trata de un comercio ya habilitado que agrega una unidad se forma el expediente consignando los datos de la unidad o unidades a incorporar (copia de dominio, tarjeta verde, carnet u otro dato relacionado al automotor). "}) |

Para exportar backup mongodump --db belaprodb

Servicies:

Create route:

POST:

http://localhost:8090/processes-api/route/

body:

{

path:string,

location: string,

name: string

}

Get routes:

GET:

http://localhost:8090/processes-api/route/

returns: [{

path:string,

location: string,

name: string

}]

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Process Service \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Get processes:

GET:

http://localhost:8090/processes-api/process

returns:[{

name:string,

description:string,

key:string

}]

Start process:

POST:

http://localhost:8090/processes-api/process/<process\_name(process.key)>/start

body: TBD

returns:{

taskId:string,

form:string

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Process Instance Service \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Get processes instances:

GET:

http://localhost:8090/processes-api/processinstance?user=nombreUser

query parameters: user: string

returns:[{

name:string,

description:string,

key:string

}]

Complete task:

POST:

http://localhost:8090/processes-api/processinstance/<processinstanceid>/nextTask

body: (depends by the process structure)

returns:{

taskId:string,

form:string

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Booking Service \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Get busy time:

GET:

http://localhost:8090/booking-api/bookings

queryparams: start:isodate,end:isodate,resourcename:string

Create appointment:

POST:

http://localhost:8090/booking-api/bookings

body:{

start:isodate,

end:isodate,

summary:string,

resourceName:string

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Resource Service \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Get resources:

GET:

http://localhost:8090/booking-api/resource/

queryparams: category:string

returns:[{

resourceName:string,

calendarpath:string,

category:string[]

}]

Create resources:

POST:

http://localhost:8090/booking-api/resource

body: {

resourceName:string,

calendarpath:string,

category:string[]

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* News Service \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Create article:

POST:

http://localhost:8090/booking-api/news/

body:

{

title:string,

body:string,

imageLink:string,

category:string[],

keywords:string[],

start:isodate,

end:isodate,

enabled:boolean,

author:string,

public:boolean,

roles:string[]

}

{"title": "Test 3",

"body": "Lorem ipsum ….",

"imageLink": "http://i.telegraph.co.uk/multimedia/archive/03137/Miss\_World\_rolene\_3137841c.jpg",

"category": [

"rentas",

"mesa"

],

"keywords": [

"pago",

"impuesto"

],

"start": "2016-02-01T00:00:00Z",

"end": "2016-02-01T23:59:59Z",

"enabled": true,

"author": "Virginia",

"public": true,

"roles": ["roleB"]

}

Get articles:

GET:

http://localhost:8090/booking-api/news/

isodate: yyyy-MM-dd'T'HH:mm:ss.SSS'Z'

por ej: 2016-02-01T01:00:00.000Z

queryparams: public:boolean,start:isodate,end:isodate,roles:string[]

returns:[{

title:string,

body:string,

imageLink:string,

category:string[],

keywords:string[],

start:isodate,

end:isodate,

enabled:boolean,

author:string,

public:boolean,

roles:string[]

}]

Get article by id:

GET:

http://localhost:8090/booking-api/news/<articleid>

returns:{

title:string,

body:string,

imageLink:string,

category:string[],

keywords:string[],

start:isodate,

end:isodate,

enabled:boolean,

author:string,

public:boolean,

roles:string[]

}

Functional:

* Web Responsive. (Emulator Chrome, iPhone & Android devices)
* News (publics/ privates, by roles, )
* Processes (query, start)
* Cross Browsing (Chrome, Firefox).

Tool: <https://app.crossbrowsertesting.com>

<http://127.0.0.1:8000/#/welcome>

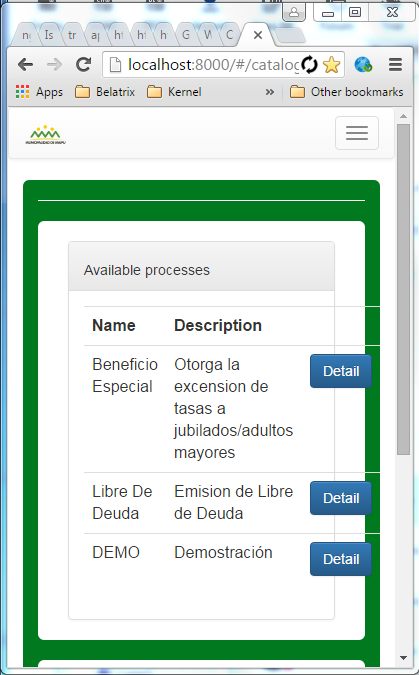
usar : <http://local:8000/#/welcome>

Automation (Protractor):

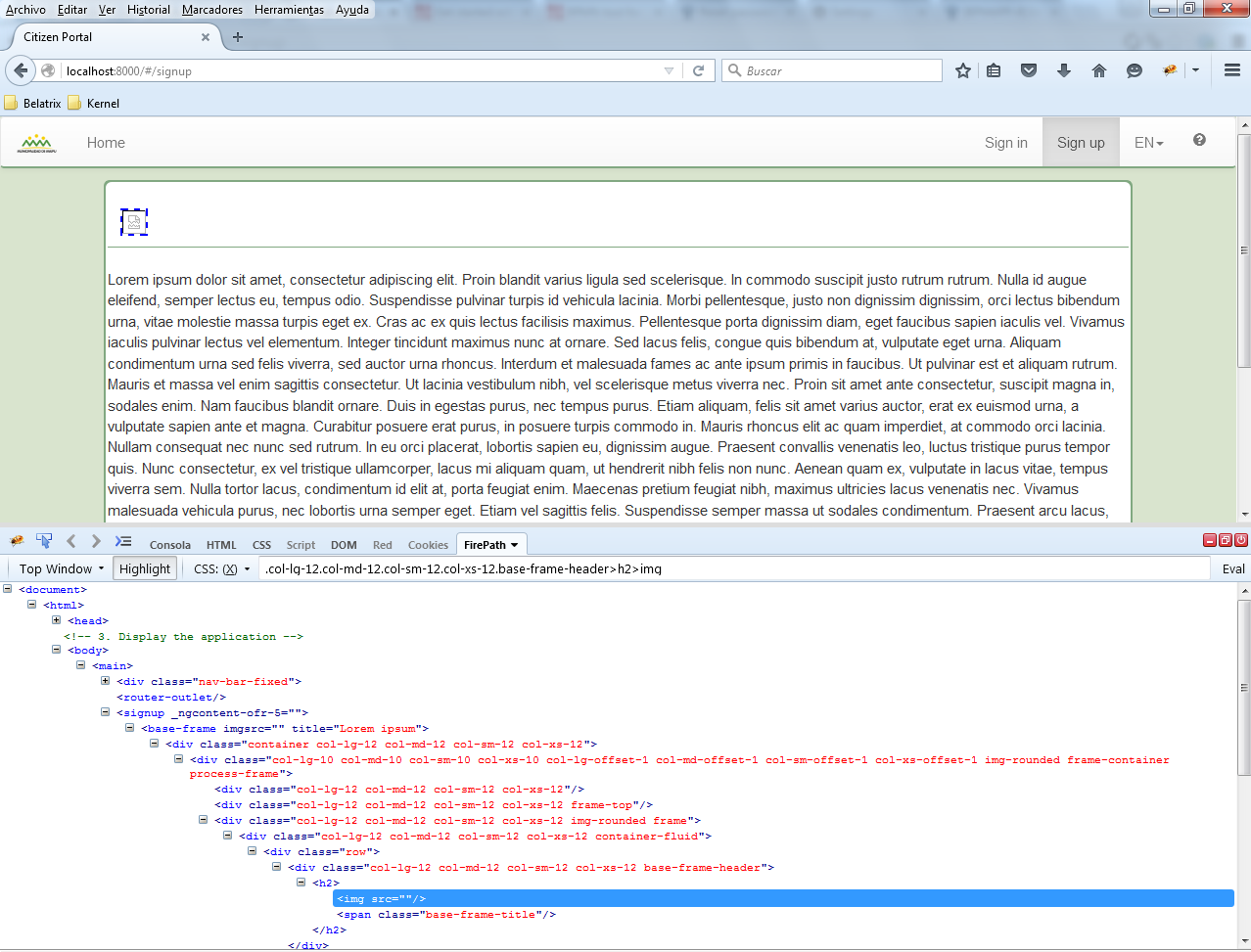
* Verify home fields
* Internationalization (home Page, login Page)
* Login (positive/negative)
* Declare pending specs

Bugs:

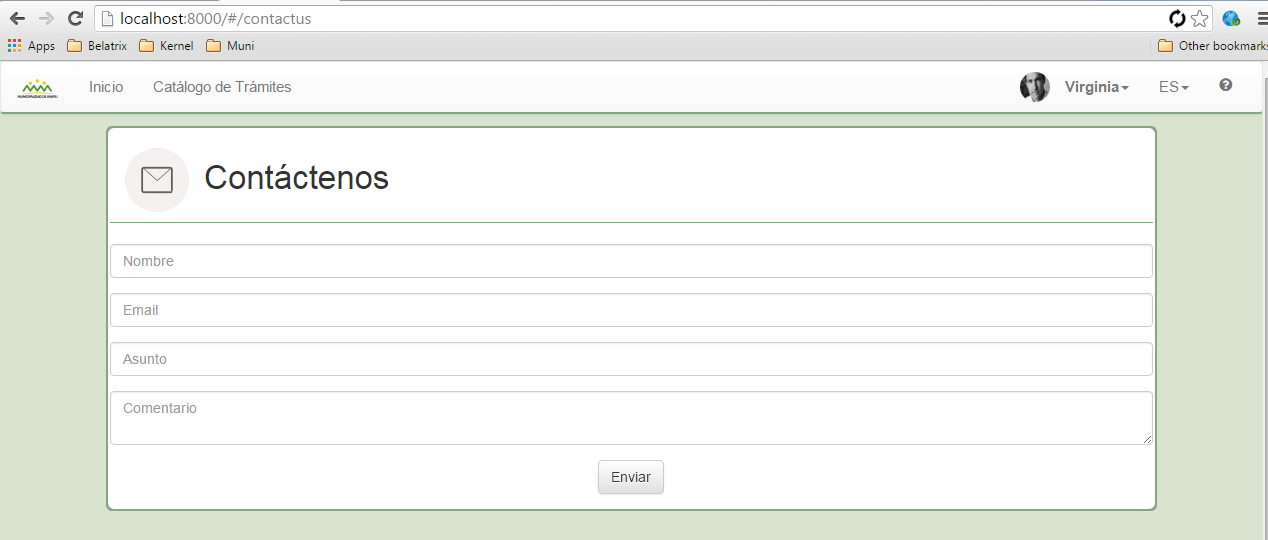
1. Web Responsive: Process Catalog overlap images and does not display the search. RESOLVED



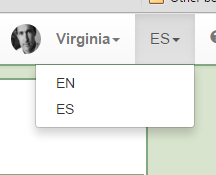
1. Modificar el mensaje “Su proceso se encuentra en la etapa e ingresó en la fecha y hora ”. RESOLVED.
2. Agregar Internacionalización para los procesos.
3. Empty image in Sign Up: see on Firefox:



1. Required Fields do not have an indicator. (\*)



1. Internationalization display the option selected. E.g. We should not display ES if we have been selected previously.



1. Municipal icon should not contain text to be more meaningful and appropriate for design (because it is too small) and internationalization (because it doesn’t change properly).

D:\Muni\bugs\logo.png

1. The app should keep the page when the idiom is changed.

* Go to contacts
* Change idiom
* See the page

Result: app redirects to home, it should keep contacts.

1. App login without username

* Clear cache
* Login with user:””, pass:”123”

1. Hamburger does not close after click on it.
2. My Profile redirects to <http://localhost:8000/#/help>. It should redirect to another page for profile.
3. On Safari doesn´t load. Displays the error: **EXCEPTION: Error during instantiation of BrowserDetails! (Token Promise<ComponentRef> -> DynamicComponentLoader -> AppViewManager -> RootRenderer -> DomRootRenderer -> AnimationBuilder -> BrowserDetails).**
4. On IEdoesn´t load.
5. ‘My Processes’ and ‘Process Catalog’ should be have the first letter in capital.
6. iPad Menu (768 x 1024 px) looks desaligned.



Deployar el proyecto DEMO:

* + - 1. Despues de hacer gulp clean (esto borra la carpeta demo), hacer gulp build:
      2. Copiar la carpeta demo de: C:\Users\vzambudio\git\appointments-extra\appointments-extra\dist\app\components
      3. A la carpeta: C:\Users\vzambudio\git\appointments-ui\dist\app\components
      4. Gulp run

Error: protractor Error while waiting for protractor to sync with the page

Solucion: Agregar: rootElement: 'div', en conf