<http://www.protractortest.org/>

http://www.protractortest.org/#/toc

Setup

Use npm to install Protractor globally with:

npm install -g protractor

This will install two command line tools, protractor and webdriver-manager. Try running protractor --version to make sure it's working.

The webdriver-manager is a helper tool to easily get an instance of a Selenium Server running. Use it to download the necessary binaries with:

webdriver-manager update

Now start up a server with:

webdriver-manager start

This will start up a Selenium Server and will output a bunch of info logs. Your Protractor test will send requests to this server to control a local browser. You can see information about the status of the server at <http://localhost:4444/wd/hub>.

todo-spec.js:

describe('angularjs homepage todo list', function() {

it('should add a todo', function() {

browser.get('https://angularjs.org');

element(by.model('todoList.todoText')).sendKeys('write first protractor test');

element(by.css('[value="add"]')).click();

var todoList = element.all(by.repeater('todo in todoList.todos'));

expect(todoList.count()).toEqual(3);

expect(todoList.get(2).getText()).toEqual('write first protractor test');

// You wrote your first test, cross it off the list

todoList.get(2).element(by.css('input')).click();

var completedAmount = element.all(by.css('.done-true'));

expect(completedAmount.count()).toEqual(2);

});

});

conf.js:

exports.config = {

seleniumAddress: 'http://localhost:4444/wd/hub',

specs: ['todo-spec.js']

};

run the test with:

protractor conf.js

suites:

|  |
| --- |
| <http://stackoverflow.com/questions/24232511/how-can-i-set-up-common-functions-that-are-available-for-my-test-suites-with-pro> |

ANGULAR 2

Conf.js

|  |
| --- |
| exports.config = {  baseUrl: 'http://http://localhost:3000/',  specs: ['spec.js'],  directConnect: true,  exclude: [],  multiCapabilities: [{  browserName: 'chrome'  }],  allScriptsTimeout: 110000,  getPageTimeout: 100000,  framework: 'jasmine2',  jasmineNodeOpts: {  isVerbose: false,  showColors: true,  includeStackTrace: false,  defaultTimeoutInterval: 400000  },  /\*\*  \* ng2 related configuration  \*  \* useAllAngular2AppRoots: tells Protractor to wait for any angular2 apps on the page instead of just the one matching  \* `rootEl`  \*  \*/  useAllAngular2AppRoots: true  }; |

Spec.js

|  |
| --- |
| describe('Protractor Demo App', function() {  var itemList=element.all(by.css('.heroes>li'));  it('should open home', function() {  browser.get('http://localhost:3000');  });    it('should have a title', function() {  expect(browser.getTitle()).toEqual('Angular 2 QuickStart');  });    it('should open Mis Tramites', function() {  itemList.get(1).click();  expect(itemList.get(1).getText()).toContain('12 Narco');  });    }); |

Angular 2 Projects:

<https://github.com/szikszail/protractor-ng-newsletter-tutorial>

Search + Contains:

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | |  | There are multiple ways to do it, e.g. with a CSS selector:  // starts-with  element(by.css("div[id^=test]"));  // ends-with  element(by.css("div[id$=test]"));  // contains  element(by.css("div[id\*=test]"));  Or, with an XPath (no ends-with here):  // starts-with  element(by.xpath("//div[starts-with(@id, 'test')]"));  // contains  element(by.xpath("//div[contains(@id, 'test')]")); | |

An expected Text:

|  |
| --- |
| var educationPage = element(by.tagName("body")).getText();  expect(educationPage).toContain("Education"); |

Verify current URL:

|  |
| --- |
| expect(browser.getCurrentUrl()).toEqual('http://localhost:8000/#/welcome'); |

//Delete all cookies

browser.driver.manage().deleteAllCookies();

Wait for URL:

|  |
| --- |
| var urlChanged = function(url) {  return function () {  return browser.getCurrentUrl().then(function(actualUrl) {  return url != actualUrl;  });  };  };  it('should click on sign in', function() {    element(by.linkText('Ingresar')).click();  browser.wait(urlChanged("http://localhost:8000/#/login"),1000);    }); |

Wait element preset: expect(pageLogin.username.isPresent()).toBeTruthy();

How to disable waiting for Angular to navigate to a page doesn’t use Angular: https://github.com/angular/protractor/blob/master/docs/timeouts.md

|  |
| --- |
| If you need to navigate to a page which does not use Angular, you can turn off waiting for Angular by setting  browser.ignoreSynchronization = true. For example:  browser.get('page-containing-angular');  navigateToVanillaPage.click();  browser.ignoreSynchronization = true;  otherButton.click();  navigateToAngularPage.click();  browser.ignoreSynchronization = false; |

Log messages: console.log(ptor.findElement(protractor.By.id('view-container')).getText());

Debug:

<https://github.com/angular/protractor/blob/master/docs/debugging.md> cmd: protractor debug conf.js

Negative comparison:

expect(false).not.toBe(true);

Pending specs:

|  |
| --- |
| describe("Pending specs", **function**() { |
| [¶](http://jasmine.github.io/2.0/introduction.html#section-13)  Any spec declared with xit is marked as pending. | xit("can be declared 'xit'", **function**() {  expect(true).toBe(false);  }); |

it("can be declared by calling 'pending' in the spec body", **function**() {

expect(true).toBe(false);

pending();

});

});

URL protractor project: <https://github.com/CarmenPopoviciu/angular2-go-protractor>

PDF con commandos: https://cli.angular.io/reference.pdf

Angular 2

Clonar proyecto

npm install

Abrir proyecto desde visual code

Para crear un componente:

Ir a la carpeta raíz del proyecto y escribir ng g component nombreComponente

**Correr las pruebas unitarias**: ng test

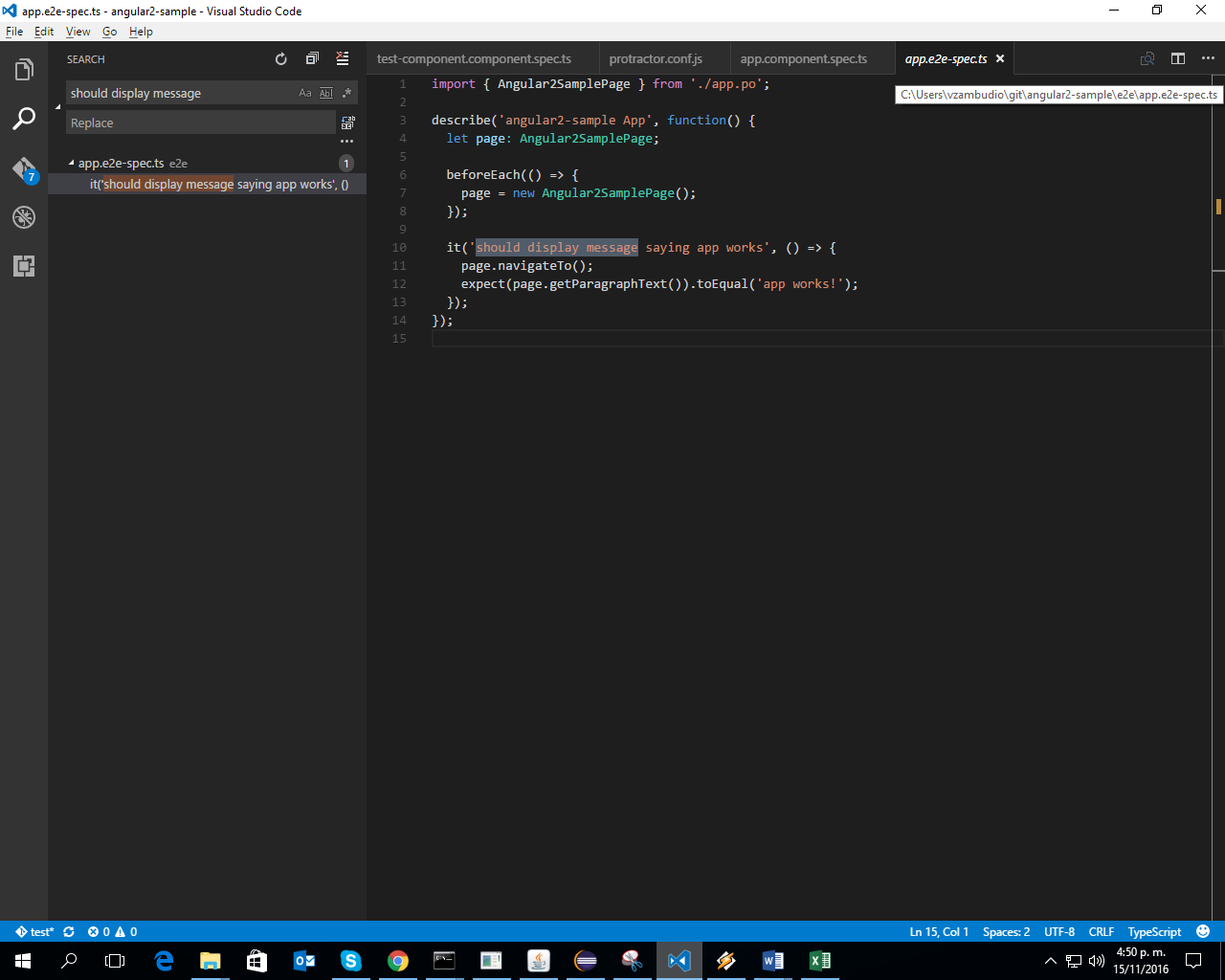
Para correrolo con protractor:

Instalar protactor: npm install -g protractor

**Correr pruebas ui:** ng e2e

Para reportar bugs npm bugs nombreProyecto

El archivo de los tests es: app.2e2-spec.ts



CAMUNDA

MAVEN Project: <https://docs.camunda.org/get-started/bpmn20/deploy/>

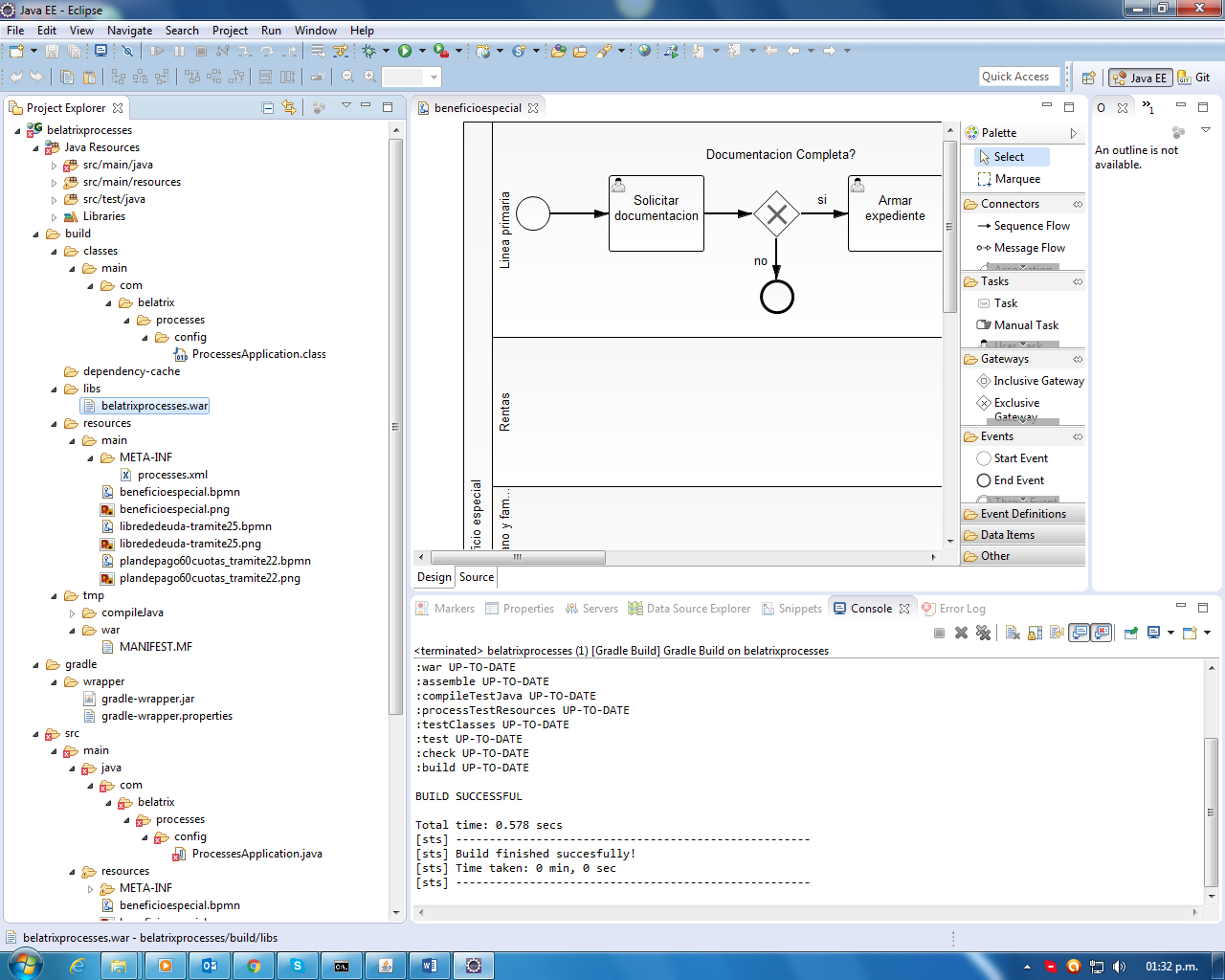
C:\Users\vzambudio\workspace-test\loan-approval\src\main\resources

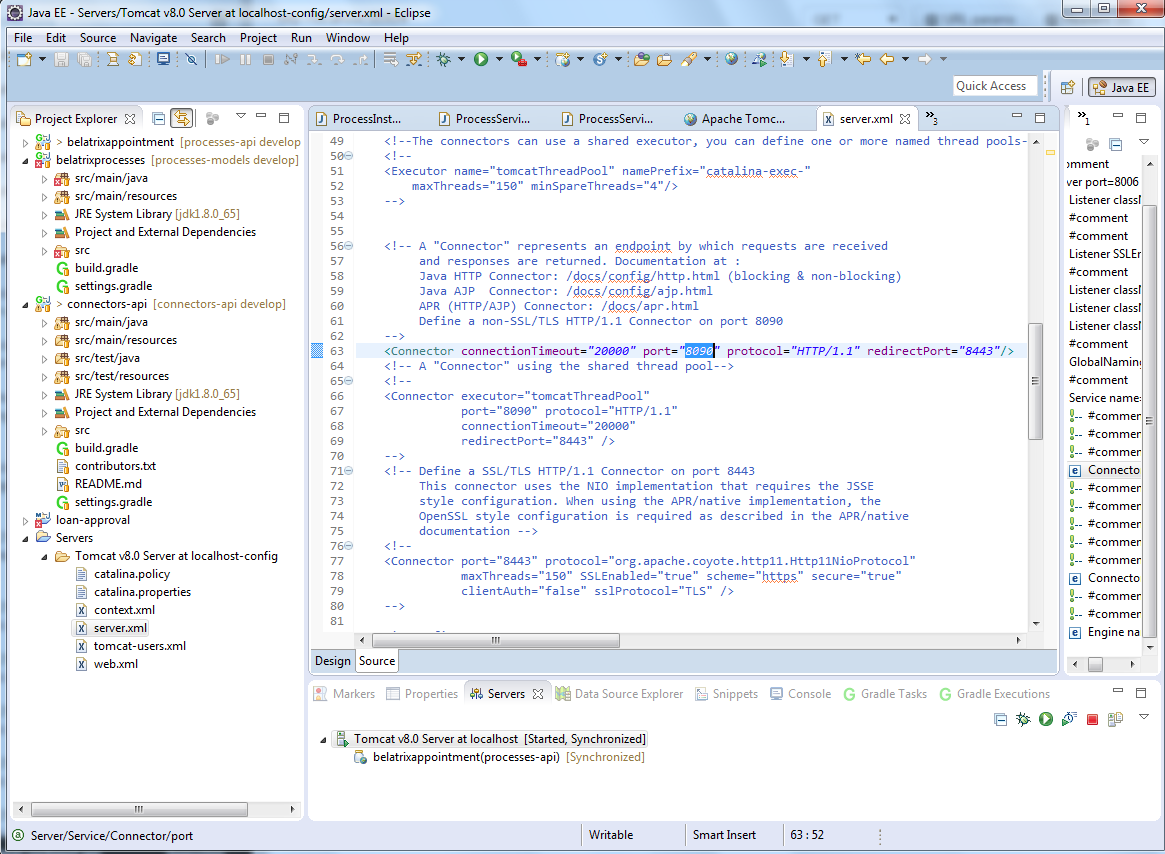
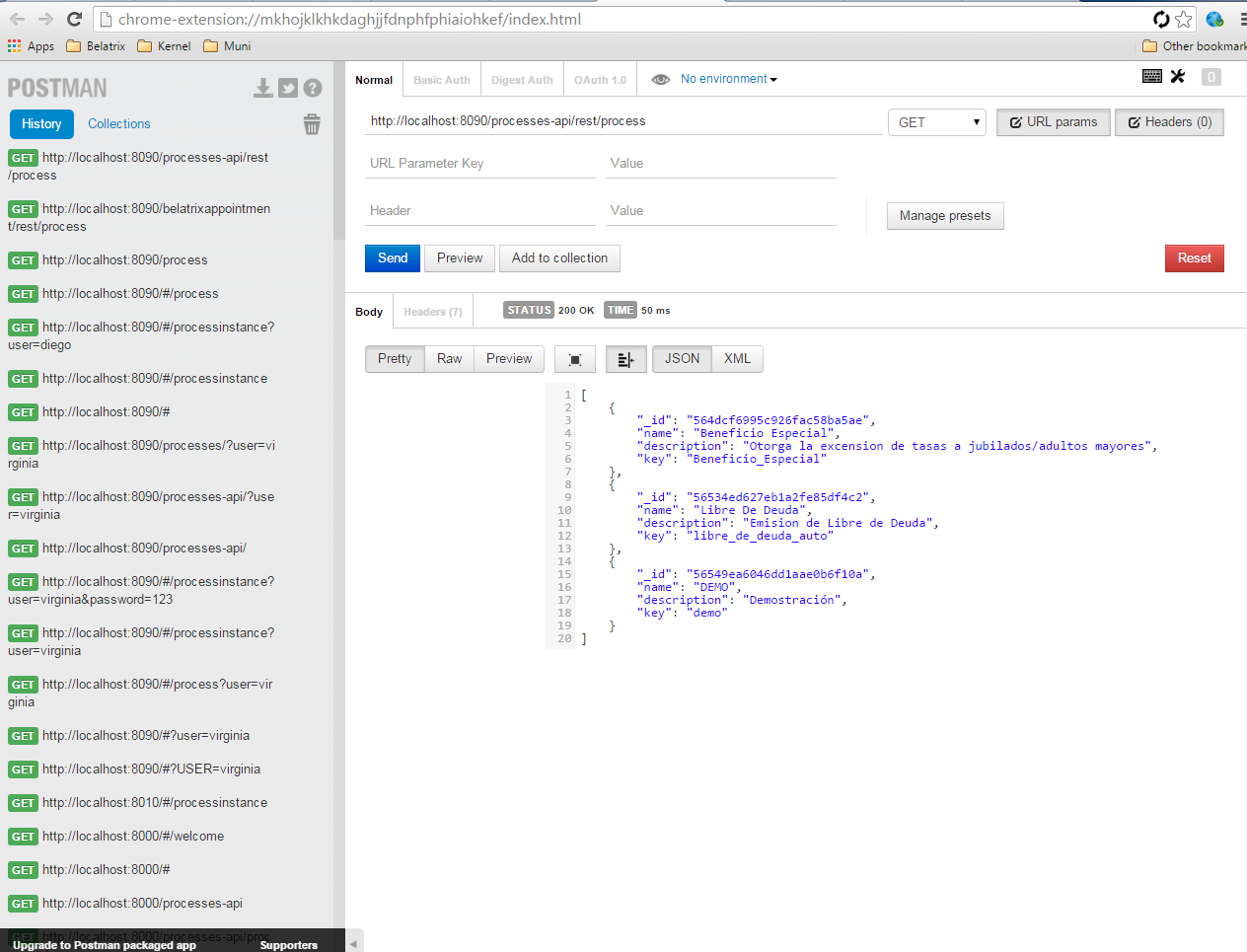
Pom.xml run as Maven install

GRADLE Project:

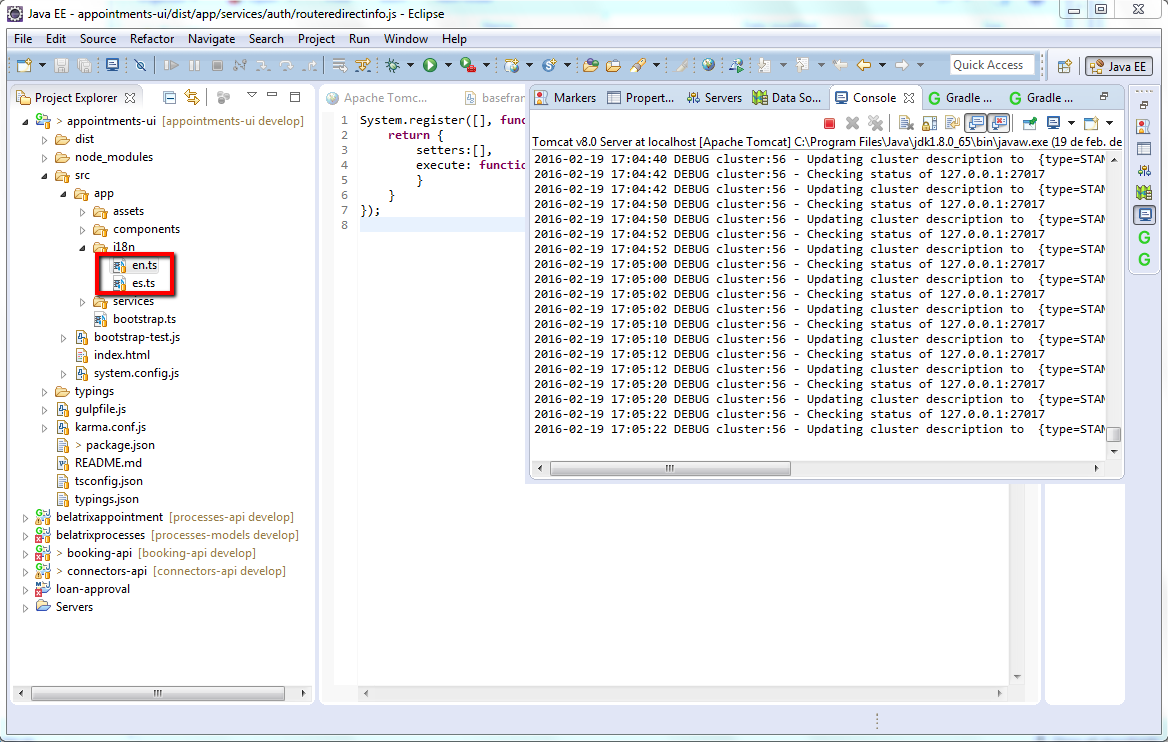
1. Abrir eclipse Mars
2. Pegar modelo bpmn en carpeta del proyecto D:\Muni\belatrixprocesses\belatrixprocesses\build\resources\main
3. Borrar war del proyecto (D:\Muni\camunda-bpm-tomcat-7.4.0\server\apache-tomcat-8.0.24\webapps)
4. Run belatrixprocesses (D:\Muni\belatrixprocesses\belatrixprocesses )as Gradle build.. escribir ‘build’ en el cuadro que aparece. Run (clean build de gradle)
5. Guardar war en carpeta del server de camunda

D:\Muni\camunda-bpm-tomcat-7.4.0\server\apache-tomcat-8.0.24\webapps



1. Start camunda.bat (arranca en el Puerto que esta configurado en el mismo archivo D:\Muni\camunda-bpm-tomcat-7.4.0)
2. Abrir cockpick (user:demo/pass:demo)
3. SI da error, ver logs: D:\Muni\camunda-bpm-tomcat-7.4.0\server\apache-tomcat-8.0.24\logs (catalina..)
4. Apuntar a :http://localhost:8090/processes-api/rest/process
5. 
6. Para levantar procesos: el connectors-api debe estar en el tomcat de eclipse. Ir a la pestaña gradle task y correr clean, eclipse y build para connectors.
7. Refrescar siempre los proyectos en eclipse después de actualizarlos desde git
8. 

Internationalization files:



Como levantar la app citizen portal: http://localhost:4200/#/welcome

