

# Elastic Workflow

Ohjelmistotuotantoprojekti, Syksy 2012

Ryhmä: Daria Antonova, Panu Klemola, Herkko Virolainen

Asiakas: Pietu Pohjalainen

Ohjaaja, Kasper Hirvikoski

Helsingin Yliopisto, 2012

[http://www.cs.helsinki.fi/group/ohtu/s-2012/aihe\\_elasticwf.html](http://www.cs.helsinki.fi/group/ohtu/s-2012/aihe_elasticwf.html)

## Yleistä

Projektin tavoitteena oli rakentaa verkossa toimiva prosessin mallinnus työkalu. Ohjelma on rakennettu ns yksisivuiseksi verkkosivuksi, eli verkkosivua ei tarvitse ladata uudelleen, vaan ohjelma tallentaa tietonsa kokoaika tietokantaan.

## Asennusohje

Ohjelma itsessään ei tarvitse minkäänlaista asennusta. Tällähetkellä se toimii ja pyörii Herokussa itsellään.

Paikallista asennusta varten tarvitaan typesafe stack asennus, jonka mukana tulee play for Scala. Tämän lisäksi lähdekoodeista(public/javascripts/app.js) pitää muuttaa domain asetus viittaamaan siihen domainiin, johon asennus tehdään. Tämän jälkeen palvelun voi käynnistää komentoriviltä menemällä ohjelman juurikansioon ja komentamalla "play run".

## Vaatimukset

### Toiminnalliset vaatimukset

Tarkat toiminnalliset vaatimukset löytyvät projektin backlogista( linkki 4). Yleisellä tasolla vaatimukset:

- Käyttäjä voi luoda elementtejä
- Käyttäjä voi muuttaa elementtejä

- Softa tallentaa elementit tietokantaan

Ei toiminnallisia vaatimuksia ohjelmalla on:

- Copypasten on toimittava tekstielementeissä.
- Useampi käyttäjä pystyy editoimaan tietoa samanaikaisesti
- Uusia elementtejä voidaan luoda järjestelmään.
- Selaimen nappien tulee toimia loogisesti.

## Arkkitehtuurikuvaus

### Backend:

Backend pyrkii noudattamaan RESTful periaatetta ja tarjoilee dataa lähinnä ajax kutsujen perusteella. Data kulkee front- ja back-endin välissä JSON muotoisena.

Backend on toteutettu Play 2.0 frameworkin päälle.

### Models:

#### 1. Table

Tietokantataulu: ei ole

Abstrakti luokka(trait) joka on kaikkien modeleiden ylliluokka.

Määrittää CRUD funktionaalisuuden copypastekoodin välttämiseksi ja vianmäärityksen helpottamiseksi. Jokainen model kutsuu näitä metodeita eri SQL tiedusteluissa.

#### 2. Model

Tietokantataulu: models

Käytetään models -tietokantataulun muokkamiseen. Model on siis on ohjelmassa käsite jonka sisällä on eri prosesseja, prosessin elementtejä sekä relaatioita elementtien välillä. Jokaisella modelilla on nimi ja luomispäivä.

#### 3. Process

Tietokantataulu: processes

Käytetään processes -tietokantataulun muokkamiseen. Jokaisella prosessilla on nimi, luomispäivä ja se voi sisältää elementtejä ja relaatioita elementtien välillä.

#### **4. ModelProcess**

Tietokantataulu: modelProcesses.

Käytetään spesifioimaan mihin prosessiin jokin tietty model kuuluu. Eli modelProcesses on siis tietokannassa oleva välitaulu, jossa on prosessien ja niihin kuuluvien modeleiden id:t.

#### **5. ProcessElement**

Tietokantataulu: processElements.

Sisältää kaikki prosessien elementtien (eli siis Startin, Endin, Swimlanen ja Gatewayn) datat sekä niiden muuttamiseen tarvittavat metodit.

#### **6.ElementType**

Tietokantataulu: elementTypes.

Koska kaikki erityyppiset elementit tallennetaan samaan tietokantatauluun joudutaan käyttämään tätä taulua, jotta erityyppiset elementit voidaan erotella toisistaan.

#### **7.Relation**

Tietokantataulu: relations

Ohjelmassa käytetyt nuolet kuvataan backendissä relaatioina elementiltä toiselle elementille.

#### **8. RelationType**

Tietokantataulu: relationTypes

Tätä modelia voitaisiin käyttää jos tarvittaisiin toisenlaisia nuolia, mutta niitä ei ehditty vielä toteuttaa projektissa.

specs2  
BDD testaus.

anorm  
Tietokanta.

routes  
Määrittelee mille kontrollerille pyynnöt ohjataan.

## Frontend:

Frontend on rakennettu Backbone.js:n päälle ja piirtämiseen käytetään Raphael.js:ää ja templatien piirtämiseen Mustache.js:ää. Ulkoasu on rakennettu Twitter bootstrapin päälle.

### Backbone models:

Backbonessa on erilliset modelit jokaiselle elementille. Niillä on ominaisuuksia kuten sijainti näytöllä ja value, vain nämä tiedot lähetetään backendille.

### Backbone Views:

Jokaisella näytöllä näkyvällä elementillä on tiedot modelissa. Modeli annetaan aina omalle view oliolle, joka osaa hoitaa modelin piirtämisen ja kuunnella siihen liittyviä tapahtumia.

### Raphael.js

Apukirjasto elementtien piirtämiseen

### Mustache.js

Templatien piirtämiseen käytetty apukirjasto

### Twitter bootstrap

Alusta ohjelmisivun rakenteelle.

### RaphaelDrawFunction.js

Apufunktioita raphaelille, jotta elementtejä voidaan liikuttaa hiirellä ja relaatiot piirtyvät oikein.

### App.js

Vastuussa ohjelman alustamisesta. Alustetaan myös globaalit taulukot, joihin osa tiedoista tallennetaan ohjelman omaa käyttöä varten.

## Toiminta

### Käyttö skenaarioita

#### Skenaario 1:

Käyttäjä haluaa katsastaa miten jokin prosessi suoritetaan.

- Vaatimukset: jokin prosessi on olemassa valmiiksi.

1. käyttäjä avaa selaimen.

2. käyttäjä menee osoitteeseen <http://morning-fjord-4117.herokuapp.com/edit> ( tai mihin domainiin ohjelma onkaan mahdollisesti asennettu)

- Tulos: käyttäjä näkee prosessin.

#### Skenaario 2:

Käyttäjä haluaa luoda uuden prosessin

-Vaatimukset: Käyttäjä on avannut sivun.

<http://morning-fjord-4117.herokuapp.com/edit>

1. Käyttäjä klikkaa New Process -nappulaa.

-Tulos: Uusi prosessi on luotu, tallennettu tietokantaan ja käyttäjä näkee sen ja sen sisältämät elementit ruudulla.

#### Skenaario 3:

Käyttäjä haluaa luoda uuden elementin

- Vaatimukset: tietokannassa on ainakin yksi malli ja prosessi.

1. Käyttäjä avaa ylävalikosta new valikon ja valitsee sieltä sopivan elementin.

#### Skenaario 4:

Käyttäjä haluaa luoda relaation kahden elementin välille.

1. Käyttäjä valitsee elementin, josta relaation pitäisi lähteä, eli toimii relaation alkupisteenä.

2. sivuun avautuvasta valikosta valitaan uusi relatio ja klikataan sitä elementtiä, johon relatio halutaan tehdä.

## Skenaario 5:

Käyttäjä haluaa muuttaa elementin nimen

1. Käyttäjä valitsee elementin, jonka nimeä haluaa muuttaa.
2. Sivulle avautuvasta valikosta valitaan tekstikenttä jossa on tämänhetkinen nimi ja kirjoitetaan siihen uusi nimi. Kun painetaan enteriä tai viedään hiiri johonkin muualle, tiedot tallentuvat.

## Käyttöönotto

Ohjelmisto vaatii play 2.0 sovelluskehiksen asennuksen koneelle.

Ohjelmisto voidaan käynnistää hakemalla githubista lokaali kopio ja käynnistämällä se komennolla play run.

Ohjelmisto on tällä hetkellä käytettävissä webin kautta osoitteessa <http://morning-fjord-4117.herokuapp.com/>. Jos ja kun ohjelmaa halutaan ajaa omalla koneella on suositeltavaa editoida kansiossa public/javascripts/app.js tiedostosta muuttuja domainHost (rivi 21). Ohjelman frontend käyttää tätä muuttujaa backendin osoitteena.

Linkit:

1. Github: <https://github.com/Herkko/ElasticWorkflow>
2. Heroku: <http://morning-fjord-4117.herokuapp.com/>
3. Jenkins: <http://jenkins.staff.cs.helsinki.fi/job/Elastflo/>
4. Project Backlog: [https://docs.google.com/spreadsheet/cc?key=0AvIhIPV1M\\_FfdFRiRkFpdDNNMlUtMmNIVU9rTVRseXc#gid=0](https://docs.google.com/spreadsheet/cc?key=0AvIhIPV1M_FfdFRiRkFpdDNNMlUtMmNIVU9rTVRseXc#gid=0)