**SPA 인수인계 확인서[[1]](#footnote-0)**

인계자 : 정현태

인수자 : 김성현

인계날짜 : 2019/01/03

**목차**

**0. 필요한 기술 스택**

1. **Client**

**1.1 클라이언트 구상도 및 개요**

**1.2 인터페이스 설명**

**1.3 Future Work**

**1.4 Java Swing Version[[2]](#footnote-1)**

1. **Server**

**2.1 서버 구상도 및 개요**

**2.2 코드 설명**

**2.3 프로그램 및 패키지 인스톨**

**2.4 방화벽 정책**

**` 2.5 서버 실행 방법**

**2.6 Future Work**

**2.7 Java Swing Version 을 위한 서버 코드**

**2.8 참고자료**

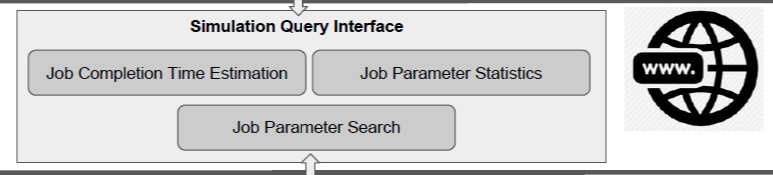
**0. 필요한 기술 스택**

SPA 프로젝트에서 서버 및 클라이언트 파트를 맡게 되었다면 해당 기술을 필요로 한다. 우선 클라이언트 파트에서는 웹페이지 기반으로 제작되어 있기에 HTML 을 다룰 줄 알아야하며 동적 로딩 및 홈페이지의 기능 구현을 위해 ajax,jquery에 대한 이해가 필요하다.

서버파트에서는 Node js 프레임워크를 이용하여 제작되었기에 javascript를 다룰 줄 알아야하며 REST 규칙을 준수하며 제작되었기에 REST 아키텍쳐에 대한 이해가 필요하다. 또한 Node js 에서 비동기화 및 동기에 대한 작업을 하기 위한 콜백함수 및 async/sync 함수 혹은 promise 함수를 사용할 수 있어야한다. api 서버 제작을 위해 사용한 express 패키지에 대한 이해가 필요하며 node js - mongodb 연결을 위한 mongoose 패키지에 대한 이해가 필요하다. Node js 서버에서 jade 란 템플릿 엔진을 사용하여 html로 변환하기에 jade 템플릿 엔진에 대한 경험 또한 필요하다. 혹시 모르는 점이 있다면 [kathor888@gmail.com](mailto:kathor888@gmail.com) 로 메일을 보내면 답장을 드리겠다.

**1. Client**

**1.1 클라이언트 구상도 및 개요**

****

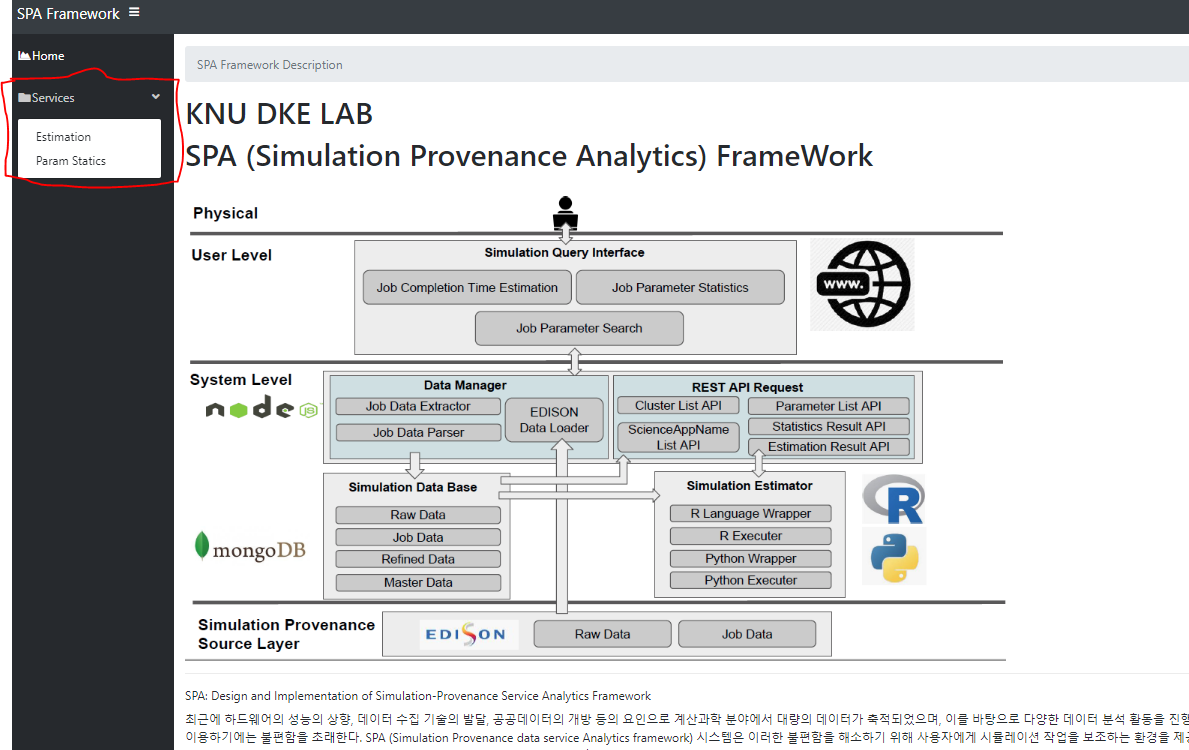
SPA Framework의 클라이언트는 사용자에게 시뮬레이션에 관한 질의 요청을 수행하는 인터페이스 역할을 하게되며 Job Completion Time Estimation, Job Parameter Statistics, Job Parameter Search 기능을 제공한다. 클라이언트는 사용자에게 웹의 형태로 제공되기 위해 HTML을 기반으로 만들어진 웹페이지이며 Node js 서버에서 Jade라는 템플릿 엔진을 이용하였다. Jade란 Node js 프레임워크에서 html을 좀 더 간단하게 사용하여 이를 express 패키지를 통해 REST API의 규칙을 준수하며 홈페이지를 사용자에게 제공해 줄 수 있다. Jade로 작성된 문서는 렌더링과정을 거쳐 HTML로 변환된 뒤 사용자에게 제공하게 된다.

**1.2 인터페이스 설명**

SPA Framework의 클라이언트는 크게 3가지 페이지로 구성되어 있다. 프로젝트에 대한 아키텍쳐와 간단한 설명이 표시되어있는 /spa (main page), 시뮬레이션 추정 시간을 서비스하는 /spa/predict , 파라미터 랭킹 서비스를 제공해주는 /spa/statistics 페이지로 구성되어있다. 예를들면 <http://155.230.34.149:3000/spa/statistics> 이런식으로 페이지 링크가 구성된다.

[[3]](#footnote-2)

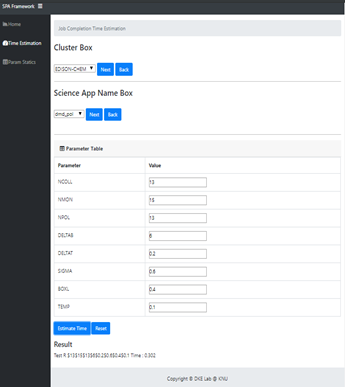
**1.2.1 Main Page**



메인 홈페이지며 아키텍쳐와 프로젝트에 대한 간단한 설명이 담겨 있다. 좌측의 Home 을 클릭시 메인페이지로 이동하며 Services를 클릭시 하위 노드가 출력되며 현재 서비스 중인 기능들을 표시한다. Estimation을 클릭시 Job Completion Time 기능을 지원하는 사이트인 /spa/predict 로 이동하며 Param Statistics 클릭시 Job Parameter Statistics 기능을 지원하는 사이트인 /spa/statistics 로 이동한다. 코드는 SPA/views/main.jade에 담겨 있다.

**1.2.2 Job Completion Time Estimation**

시뮬레이션 시간 추정 서비스는 SPA 시스템내의 Simulation Query Interface 중 Job Completion Time 인터페이스를 통해 서비스를 제공한다. 시뮬레이션 시간 추정 서비스에는 Request Handler에서 Cluster API List, ScienceAppName List, Parameter List, Estimation Result API 가 사용된다. 다음과 같이 현재 구현이 되어 있다.

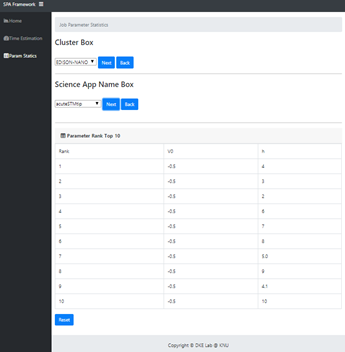


SPA 시스템에서 제공하는 시뮬레이션 시간 추정 서비스의 View

그림 에서 Cluster Box에는 Cluster List API로부터 제공받은 EDISON 플랫폼의 Simulation 소프트웨어의 종류가 리스트에 출력된다. 그후 Cluster Box List 의 Next 버튼을 클릭 시 Cluster에 해당하는 ScienceAppName List가 ScienceAppName List API로부터 제공받아 리스트에 종류가 출력된다. 원하는 ScienceAppName을 입력한 뒤 Science App Name Box의 Next 버튼을 클릭 시 ScienceAppName에 해당하는 Parameter Set이 Parameter List API로부터 제공받아 테이블의 형태로 Parameter셋과 value를 입력할 input box가 제공된다. 원하는 Value를 입력한 뒤 Estimation Time 버튼을 클릭 시 Estimation Result API가 호출되어 Input Data를 서버에 전송 후 R Wrapper, R Executor을 통해 수행 시간을 추정 후 결과 값을 하단의 Result 라벨에 출력하게 된다. Reset 버튼을 누르면 모든 상태 값들이 초기화되며, 각 Box에 해당하는 Back 버튼을 클릭 시 이전 상태로 돌아가게 된다. 코드는 SPA/views/predict.jade에 담겨 있다. 사용된 api의 경우 SPA/apps/api/api\_bunch.js 에 담겨 있으며 코멘트를 통해 표시하였다. 사용된 웹페이지 백그라운드 로직의 경우 SPA/src/fun.js에 담겨 있으며 함수명을 통해 표시하였다.

**1.2.3 Job Parameter Statistics**

파라미터 랭킹 서비스는 SPA 시스템내의 Simulation Query Interface 중 Job Parameter Statistics 인터페이스를 통해 서비스를 제공한다. 파라미터 랭킹 시스템 서비스에는 Request Handler에서 Cluster API List, ScienceAppName List, Statistics Result API 가 사용된다. 다음과 같이 현재 구현이 되어 있다.



SPA 시스템에서 제공하는 파라미터 랭킹 서비스의 View

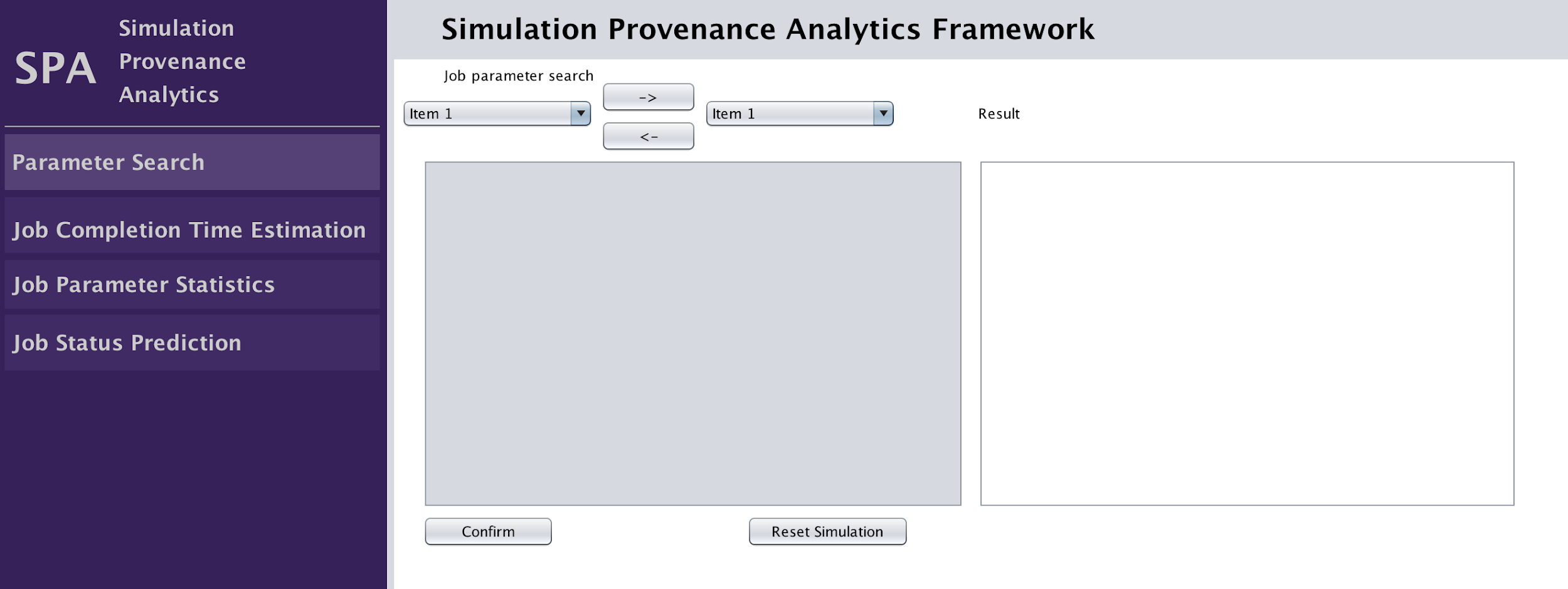
그림에서 Cluster Box 와 Science App Name Box까지의 시나리오는 시뮬레이션 추정 서비스와 동일한 로직을 사용한다. Science App Name Box의 Next 를 클릭 시 Input Data를 기반으로 Statistics Result API 호출하여 사용자가 선택한 시뮬레이션에 대해 많이 사용된 Parameter Set Rank 10을 테이블의 형태로 제공한다. Reset 및 Back 클릭 시 일어나는 시나리오 또한 시뮬레이션 추정 서비스와 동일한 로직을 사용한다. 사용된 api의 경우 SPA/apps/api/api\_bunch.js 에 담겨 있으며 코멘트를 통해 표시하였다. 사용된 웹페이지 백그라운드 로직의 경우 SPA/src/fun.js에 담겨 있으며 함수명을 통해 표시하였다.

**1.3 Future Work**

1. Description Page 즉 Main page에 내용 추가
2. 서버에 api 서비스 추가에 따른 페이지 추가

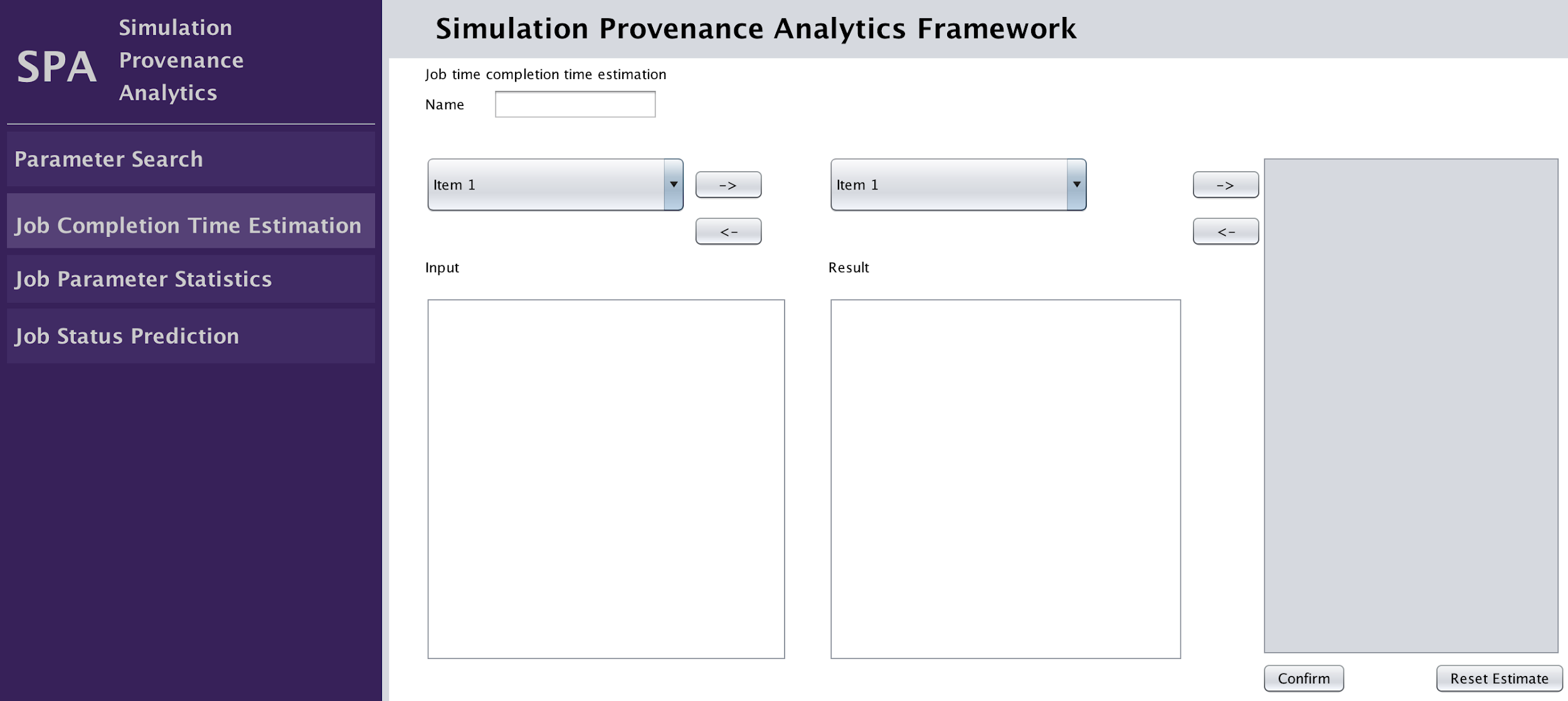
**1.4 Java Swing Version**

**1.4.1 Job Parameter Search**

****

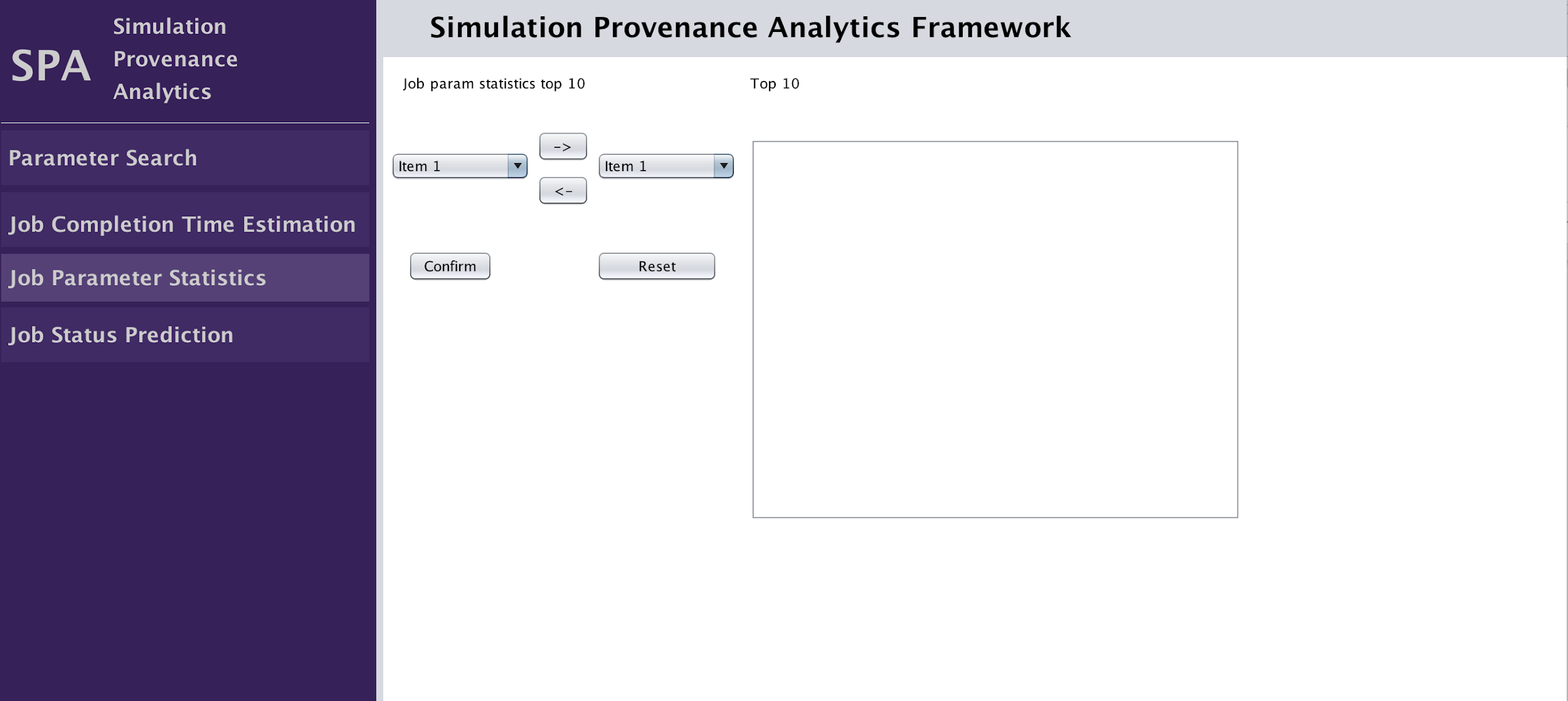
item 1에서 cluster item 2 에선 scienceAppName [[4]](#footnote-3)을 설정하면 밑 jtable 에 그와 관련된 파라미터 값을 지정할 수 있는 창이 뜬 뒤 특정 값을 지정하고 confirm 버튼을 누르면 서버에서 설정한 파라미터를 사용한 실험들을 Result 창에 보여주는 기능을 한다. Reset 버튼을 누를 시 현재패널의 모든 셋팅이 초기화된다.

**1.4.2 Job Completion Time Estimation**

****

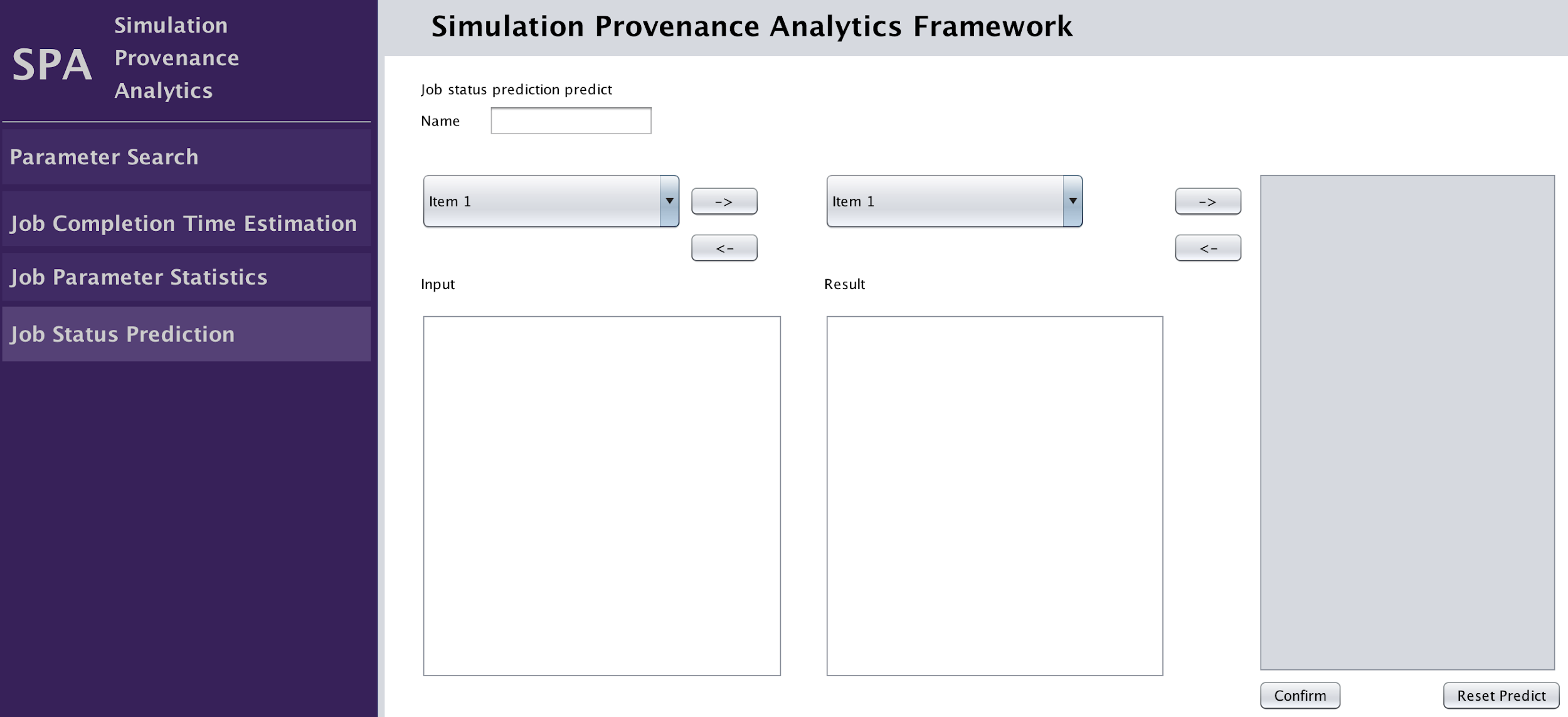
item 1에서 cluster item 2 에선 scienceAppname 을 설정하면 제일 오른쪽 jtable에 그와 관련된 파라미터 값을 지정할 수 있는 창이 뜬 뒤 특정 값을 지정하고 confirm 버튼을 누르면 서버에서 Estimation Time을 Result 창에 보여주는 기능을 한다. Reset 버튼을 누를 시 현재패널의 모든 셋팅이 초기화된다.

**1.4.3 Job Parameter Statistics**

****

item 1에서 cluster item 2 에선 scienceAppname 을 설정한 뒤 confirm 버튼을 누르면 서버에서 이 실험 셋팅에 대해 가장 많이 사용된 parameters top10을 오른쪽 test area에 보여주는 기능을 한다. Reset 버튼을 누를 시 현재패널의 모든 셋팅이 초기화된다.

**1.4.4 Job Status Prediction**

****

item 1에서 cluster item 2 에선 scienceAppName 을 설정하면 제일 오른쪽 jtable에 그와 관련된 파라미터 값을 지정할 수 있는 창이 뜬 뒤 특정 값을 지정하고 confirm 버튼을 누르면 서버에서 Job Status (성공률)을 Result 창에 보여주는 기능을 한다. Reset 버튼을 누를 시 현재패널의 모든 셋팅이 초기화된다.

**2. Server**

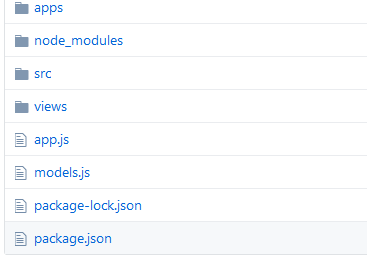
**2.1 서버 구성도 및 개요**



SPA 프로젝트에 사용된 서버는 Rest API 를 사용하는 웹서버이다. 따라서 클라리언트와 서버는 HTTP라는 규칙을 이용해서 서로 통신하게 된다. 서버의 주 기능은 API 제공과 Mongo DB 와의 연결과 외부 mysql 과의 연결이다. 우선 SPA 의 서버 및 데이터베이스는 rhel 7.4 버전에서 구동된다. 서버 프로그램으론 Node js를 사용하여 구현하였으며 Data Base 는 No-sql DB 인 Mongo DB 를 사용한다. Node js 는 v8 버전, Mongo DB 는 v4.01 두 프로그램 모두 community edition을 사용하고있다. 현재 dkelab 서버 컴퓨터에 올라가 있는 서버 프로그램 같은 경우는 별도의 설치 작업 없이 현재 실행이 가능하다.

**2.2 코드 설명**

코드에 대한 자세한 설명은 실제 소스 코드에 코멘트를 작성하였다. 허나 실제 인수인계 뒤 코딩을 하기 위해선 인수인계서에서 요구하는 기술 스택에 대한 이해가 필요하다.



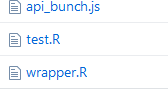
apps : api 및 데이터베이스 연동과 관련된 코드가 있다.

node\_modules : node js 프레임워크에 설치된 패키지들이 저장되어 있다.

src : css, ajax,jquery, img 등 웹페이지 제작에 필요한 리소스들이 저장되어 있다.

views : 실제 jade(html) 코드가 저장되어 있다.

app.js : 서버코드의 메인파트이다. models.js : 몽고디비와 연결시 사용할 모델들을 정의한다. package \* : 현재 프로젝트에 대한 description이 기술되어 있다.



apps/api 폴더에 들어 있는 소스 코드들이다.

api\_bunch.js : rest api 코드들이 저장되어 잇다.

test.R : R 스크립트에서 실제 기능을 하는 코드가 들어갈 곳이다.

wrapper.R : test.R과 같이 실제 기능을 하는 코드와 node js를 이어주는 인터페이스 input,output 용 코드이다.



apps/data\_control 폴더에 들어 있는 소스 코드들이다.

load\_edison.js : mysql에 저장되어 있는 edison 데이터를 서버에 저장하는 코드이다.

parser.js : edison 데이터를 파싱하여 불필요한 구문자 및 데이터를 정제하여 다시 서버에 저장하는 코드이다.

user.controller.js : api를 외부화 시킬때 사용하는 코드이며 현재 서버에서는 사용하지 않는다.

**2.3 프로그램 및 패키지 인스톨** [[5]](#footnote-4)

모든 프로그램은 rhel 7.4 에서의 설치를 가정하고 있다.

**2.3.1 Node js 설치**

1. 터미널 실행 후 **$curl --silent --location** [**https://rpm.nodesource.com/setup\_8.x**](https://rpm.nodesource.com/setup_8.x) **| sudo bash -**
2. **$sudo yum -y install nodejs**
3. **$node --version** 을 통해 설치작업이 완료되었는지 확인

**2.3.2 Mongo DB 설치**

1. **/etc/yum.repos.d/mongodb-org-4.0.repo** 를 만들고 난 뒤 아래와 같이 내용을 **수정**한다.

[mongodb-org-4.0]

name=MongoDB Repository

baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-org/4.0/x86\_64/

gpgcheck=1

enabled=1

gpgkey=<https://www.mongodb.org/static/pgp/server-4.0.asc>

2) 터미널 실행 후 **$sudo yum install -y mongodb-org**

3) **$mongo --version** 을 통해 설치작업이 완료되었는지 확인

4) **$systemctl enable mongod** 를 통해 mongod 등록

**2.3.3 Node js 패키지 설치** [[6]](#footnote-5)

1. **express 패키지 설치** : $sudo install express 사용하는 이유 : 프레임워크 사용을 통한 빠르고 효율적인 서버 개발을 위함
2. **body-parser 패키지 설치** : $sudo npm i body-parser --save 사용하는 이유 : express 의 요청 바디의 데이터에 접근하기 위해 사용
3. **mocha 패키지 설치** : $sudo npm i mocha --save-dev 사용하는 이유 : Node js 서버를 테스팅 하기 위한 패키지
4. **mysql 패키지 설치** : $sudo npm install mysql 사용하는 이유 : mysql - Node js 상호작용 관련 모듈을 사용하기 위함 일종의 드라이브이다.
5. **mongodb 패키지 설치** : $sudo npm install mongodb 사용하는 이유 : mongodb - Node js 상호작용 관련 모듈을 사용하기 위함 일종의 드라이브이다.
6. **mongoose 패키지 설치** : $sudo npm install mongoose 사용하는 이유 : Object mapper 로써 mongodb sql 을 모르더라도 javascript 문법만으로 mongodb를 사용하기 위함
7. **jade 패키지 설치** : $sudo npm install jade 사용하는 이유 : Node js에서 html 을 렌더링 하기 위해 필요한 템플릿 엔진이며 기존의 html 에 비해 간결한 문법 사용이 가능하며, express 패키지와 연동하여 api 형태로 홈페이지를 제공해 줄 수 있음

**2.4 방화벽 정책**

**2.4.1 firewalld 설치**

1. 터미널 실행 후 **$yum install firewalld**
2. **$systemctl start firewalld**
3. **$systemctl enable firewalld**

**2.4.2 port 추가**

1. **$firewall-cmd --permanent --zone=public --add-port=PORT/tcp** (PORT 에는 포트넘버기입 ex 8000)[[7]](#footnote-6)
2. **$firewall-cmd --reload** 로 서비스를 재실행하면 반영된다.

**2.5 서버 실행 방법**[[8]](#footnote-7)

1. 터미널 실행 후 **$cd SPA\_Server**
2. **$systemctl start mongod** 를 통해 MongoDB 백그라운드 실행
3. **$node app** 서버가 실행되고 콘솔에 서버가 실행되었다는 문구가 나타난다.
4. 새로운 터미널을 생성한 뒤 **$mongo** 를 타이핑 후 mongo shell 에 접속

**2.6 Future Work**

1. 테스트 툴을 통한 **서버 테스트 작업**
2. 데이터 로딩 , 파싱 코드를 모듈화는 시켰으나 이를 서버 콘솔에서 제어할 수 있는 기능 추가

**2.7 Java Swing Version 을 위한 서버 코드**[[9]](#footnote-8)

만약 Java Swing 을 이용하여 Rest api 서버와 통신을 원할 경우 다음의 클래스 설명을 꼭 읽기 바란다. Rest api 는 일종의 api 설계 방법일뿐, 결국 http를 이용한다. 이 점을 잘 숙지하길 바란다.

1. **Alldata.java** : 서버에 보낼 데이터를 AllData 에 정의하고 저장하면 된다. REST 클래스의 postinfo(Alldata topost); 에 인자로 사용하면 된다.
2. **REST.java** : SPA 클라이언트와 같은 Java application과 Node js 서버의 통신을 위해 만든 클래스이다. getInfo(String name) 와 postinfo(Alldata topost) 메소드로 이루어져있다. getInfo의 경우 서버에서 리퀘스트를 보내 원하는 정보(현재는 name 으로 구분) 을 얻어오는 기능이며, postinfo의 경우 Alldata 에 저장된 정보를 서버로 전송하게 된다. 현재 서버가 REST api 를 사용하는 웹서버임으로 http 통신을 사용하고 있다. REST.java 클래스에는 jackson lib 를 사용하여 json converting 기능을 사용하고 있다.

**2.8 참고자료**

<https://www.mkyong.com/java/how-to-send-http-request-getpost-in-java/> -- java http 통신

<http://webframeworks.kr/tutorials/nodejs/api-server-by-nodejs-01/#tocAnchor-1-8> -- node js 를 이용한 rest api server 제작방법

<http://bcho.tistory.com/1096> -- moongose 사용방법

<https://nodejs.org/ko/download/package-manager/> -- 2.3.1 Node js

<https://docs.mongodb.com/manual/tutorial/install-mongodb-on-red-hat/> -- 2.3.2 Mongo DB

<https://www.lesstif.com/pages/viewpage.action?pageId=22053128> -- 2.4 방화벽정책

1. <https://github.com/BevoLEt/SPA_0> 에 SPA 프로젝트관련 코드 ,문서, README가 정리되어있다.. [↑](#footnote-ref-0)
2. Java 버전은 참고만 하도록 한다. [↑](#footnote-ref-1)
3. CSS의 경우 <https://startbootstrap.com/template-overviews/sb-admin/> 를 이용하였다. [↑](#footnote-ref-2)
4. EDISON-CFD 2D\_Comp\_P EDISON-CFD = cluster 2D\_Comp\_P = scienceAppName [↑](#footnote-ref-3)
5. 되도록이면 terminal 에서 명령어를 통한 설치를 권장한다. [↑](#footnote-ref-4)
6. node js 패키지 설치 시 npm 을 사용한다. npm 커맨드가 동작하지 않는다면 node js 설치가 제대로 안되었다는 뜻이니 재설치를 권장한다. [↑](#footnote-ref-5)
7. 현재 SPA 서버는 포트 3000을 사용 하고 있다. [↑](#footnote-ref-6)
8. dkelab 연구실의 경우 workstation 에 SPA(or EXP) vm 에서 실행하면 된다. (pw는 관리자에게 물어보도록..) [↑](#footnote-ref-7)
9. 서버에서 사용될 javascript 언어로 된 코드 설명과 클라이언트, Excon에 사용될 java 언어로 된 코드 설명을 포함한다. [↑](#footnote-ref-8)