**“AIBRIL HR for recruit”**

**(Developer’s Guide)**

**Ver. 0.75**

**2019년 10월**

**SK주식회사 C&C**

목차

[**Revision History** 3](#_Toc23435211)

[**문서의 목적** 4](#_Toc23435212)

[**1.** **개요** 5](#_Toc23435213)

[1.1 소개 5](#_Toc23435214)

[1.2 구성 요소 7](#_Toc23435215)

[1.3 주요 Feature 8](#_Toc23435216)

[**2.** **시스템 구성** 10](#_Toc23435217)

[2.1 어플리케이션 12](#_Toc23435218)

[2.2 평가 엔진 12](#_Toc23435219)

[2.3 평가모델 생성, 학습 14](#_Toc23435220)

[2.4 점수 산정(scoring) 22](#_Toc23435221)

[2.5 키워드(Keywords), 요약문(Summarization) 23](#_Toc23435222)

[**3.** **어플리케이션 (Application)** 24](#_Toc23435223)

[3.1 구성 25](#_Toc23435224)

[3.2 주요 Sequence 27](#_Toc23435225)

[3.3 레퍼런스(Reference) 34](#_Toc23435226)

[**4.** **평가 엔진** 46](#_Toc23435227)

[4.1 구성 46](#_Toc23435228)

[4.2 주요 Sequence 47](#_Toc23435229)

[4.3 레퍼런스(Reference) 50](#_Toc23435230)

[**5.** **데이터베이스(DB)** 53](#_Toc23435231)

[5.1 테이블정의서 53](#_Toc23435232)

[5.2 저장 프로시져(Stored Procedure) 54](#_Toc23435233)

[5.3 ERD (Entity Relationship Diagram) 54](#_Toc23435234)

[**6** **개발 환경 구성** 56](#_Toc23435235)

[6.1 시스템 사양 56](#_Toc23435236)

[6.2 어플리케이션 56](#_Toc23435237)

[6.3 평가 엔진 59](#_Toc23435238)

[6.4 데이터베이스(Database) 설치 60](#_Toc23435239)

[6.5 어플리케이션 실행 64](#_Toc23435244)

[**7.** **Customization** 66](#_Toc23435245)

[**8.** **문장단위평가모델 개발도구(tool)** 67](#_Toc23435246)

[8.1 개발 배경 67](#_Toc23435247)

[8.2 기능 68](#_Toc23435248)

[8.3 실행 69](#_Toc23435249)

[**9.** **문서단위평가모델 개발도구(tool)** 74](#_Toc23435250)

[9.1 개발 배경 75](#_Toc23435251)

[9.2 기능 75](#_Toc23435252)

[9.3 실행 77](#_Toc23435253)

# **Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **버전** | **날짜** | **작성자** | **내용** |
| v0.47 | 18년 5월 | 김슬기  이재영  조형준 | Draft 작성 |
| V0.74 | 19년 10월 | 조형준 | “1.개요” ,”2.시스템 구성” 업데이트  - UI/UX 개선 사항, 평가엔진 수정 사항 반영 |
| v0.76 | 19년 10월 | 순지은  김태경  이도경 | “3. 어플리케이션”, “6. 개발 환경 구성” 업데이트  - 어플리케이션 및 개발 환경 구성 수정 사항 반영 |
| V0.77 | 19년 10월 | 조형준 | Typo 수정 등 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# **문서의 목적**

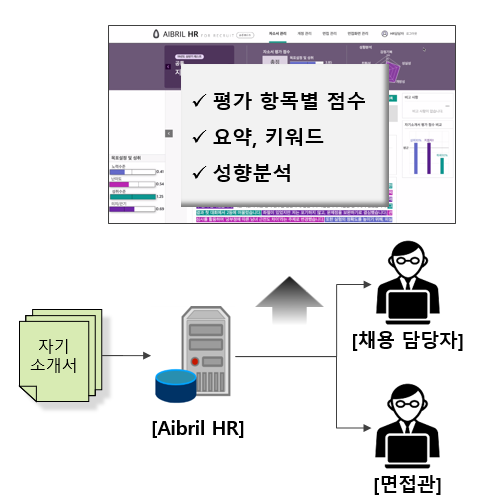
본 문서는 “AIBRIL HR for recruit”의 시스템 구성 및 주요 구성 요소에 대한 기술적인 설명을 제공합니다. 주요 내용은 아래와 같습니다.

* “AIBRIL HR for recruit”의 소개, 구성 요소에 대한 아키텍처(Architecture) 및 기술적 설명 제공
* “AIBRIL HR for recruit”를 활용한 프로젝트 수행 시 필요한 소스 구조, 주요 Use Case에 대한 Sequence Diagram, 주요 API에 대한 설명 제공
* 개발 환경 구성 및 Customization에 대한 가이드 제공
* “문장단위평가모델 생성 도구(tool)”, “문서단위평가모델 생성 도구(tool)”에 대한 설명, 사용 가이드 제공

# **개요**

“AIBRIL HR for recruit”의 소개, 구성요소, 주요 기능에 대해 설명합니다.

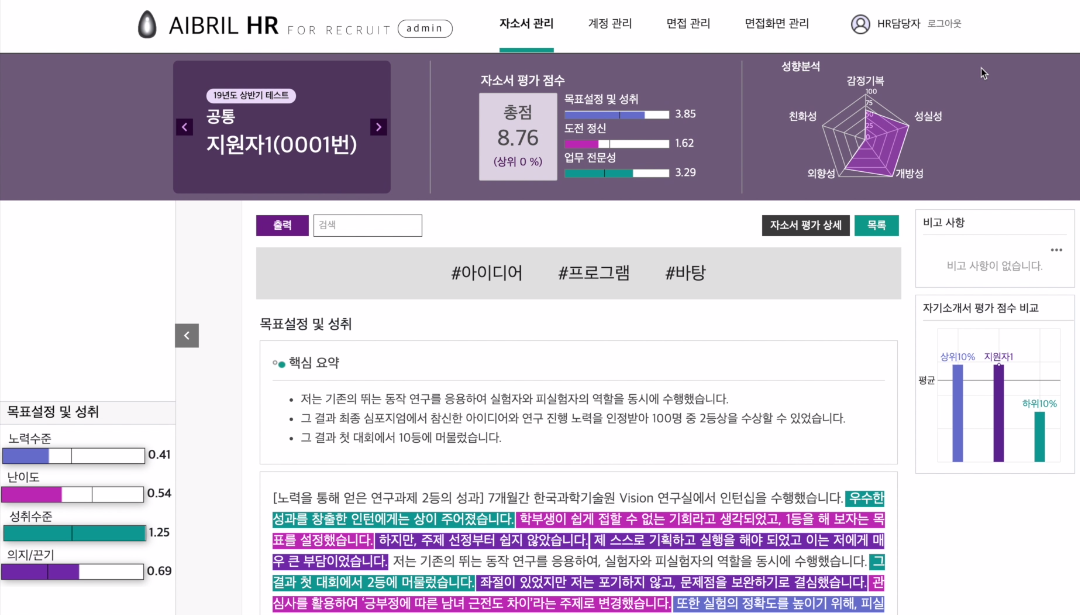
# 소개

“AIBRIL HR for recruit”는 채용 심사 시, 서류 심사에 해당되는 자기소개서 평가(scoring)을 자동으로 수행하는 솔루션 입니다.

[그림 1] 개념도

솔루션이 제공하는 주요 기능에 대한 개요는 [그림 1]과 같습니다. 자기소개서 평가 결과 외에 요약(summarization), 키워드(keyword), 지원자성향분석 결과를 제공함으로써 효과적인 면접 전형이 가능하도록 보조해 주는 기능도 함께 제공합니다.

솔루션을 사용하는 최종 사용자(end user)는 서류전형을 진행하는 채용 담당자나 면접전형을 진행하는 면접관 입니다.



5

3

4

2

1

[그림 2] 평가 결과

평가결과는 [그림 2]와 같은 형태로 제공되고, 화면구성에 대한 설명은 아래와 같습니다.

[1] 평가 항목 별 점수: 총점, 항목별 점수 표시

[2] 지원자 성향 분석 결과: 자기소개서에 기반한 지원자의 성향 분석 결과

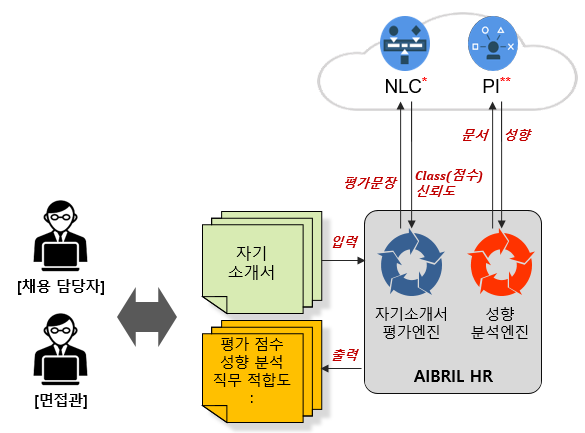
[3] 키워드(keyword) : 자기소개서에 자주 언급되는 주요 단어, 표현

[4] 요약: 자기소개서의 주요 3문장을 추출

[5] 평가 항목별 상세 분석: 항목별 점수, 평가에 사용되는 근거 문장

# 구성 요소

시스템은 [그림 3]과 같이 어플리케이션과 평가엔진으로 구성되어 있습니다.



(\*NLC: Natural Language Classifier, \*\*PI: Personality Insight)

[그림 3] AIBRIL HR for recruit 구성

어플리케이션은 사용자 UI(화면), 평가엔진 연동 기능을 담당합니다. 평가엔진은 Watson NLC(Natural Language Classifier), PI(Personality Insight) API를 이용하여 자기소개서를 점수화(scoring)하고 지원자의 성향을 분석합니다. 평가엔진은 정확한 평가점수 산정을 위해 Watson API 외에도 다양한 오픈소스 라이브러리를 함께 사용합니다.

# 주요 Feature

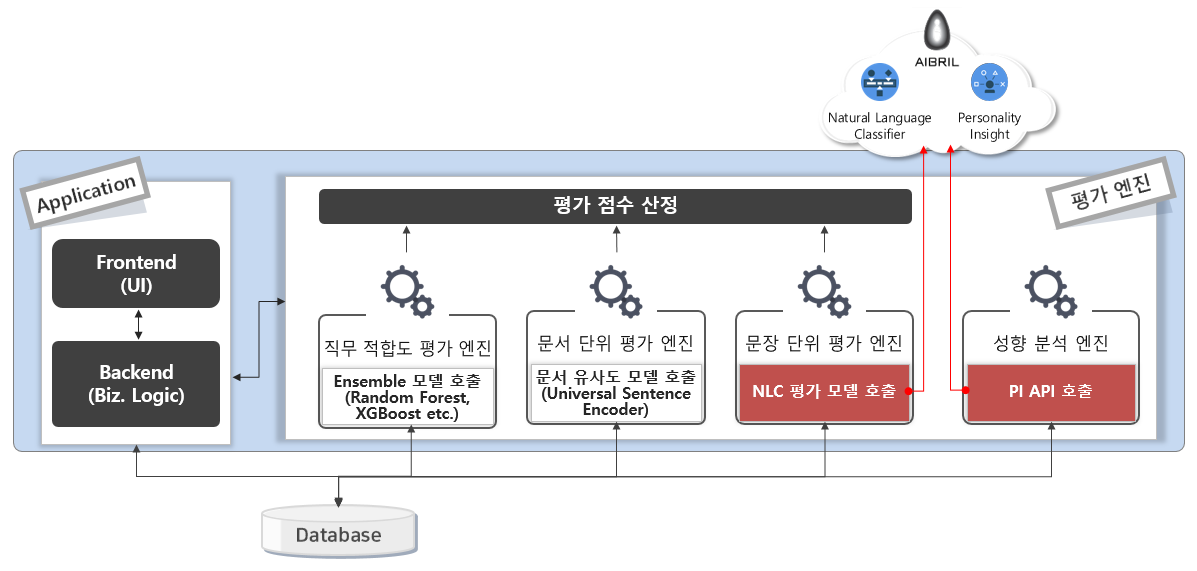
솔루션에서 제공하는 주요 Feature는 [표 1]과 같습니다.

|  |  |  |
| --- | --- | --- |
| **구분** | | **설명** |
| 시스템 기능 | 평가데이터 로드 | 데이터베이스로부터 평가결과 및 평가대상 데이터를 읽어 들입니다. |
| 평가 실행 | 평가 대상 데이터를 선택하고 평가를 실행합니다. |
| 평가 결과 | 평가 결과를 항목별로 수치화해서 제공하고 평가에 활용된 근거 문장을 제시합니다. |
| 평가 결과 검색 | 지원자 번호, 이름으로 평가 결과를 검색합니다. |
| 평가결과 다운로드 | 평가 결과를 엑셀파일로 다운로드 합니다. |
| 키워드(keyword) | 자기소개서에 자주 나타난 단어를 표시 합니다. |
| 요약문 추출 | 자기소개서의 주요 3문장을 표시 합니다. |
| 개인성향분석결과 | 자기소개서에 근거하여 분석된 개인 성향을 Spider Chart 형태로 제공합니다. |
| 설정 | 평가항목 이름, 면접 질문 등을 설정할 수 있는 메뉴를 제공합니다. |
| 계정관리 | 사용자 계정 생성, 수정, 삭제 기능을 제공합니다. |
| 평가엔진 | 문장 단위 평가 | 지원 번호 또는 이름을 이용하여 평가결과를 검색할 수 있는 검색 기능을 제공합니다. |
| 문서 단위 평가 | 평가 결과를 엑셀 파일로 다운로드 합니다. |
| 직무 적합도 평가 | 입력된 자기소개서와 지원한 업무와의 적합도를 평가하여 점수화(Scoring) 합니다. |
| 개인 성향 분석 | 자기소개서에 근거하여 개인 성향을 분석합니다. |
| 키워드(Keyword), 요약문 추출 | 자기소개서 내의 키워드, 주요 3개 문장을 추출 합니다. |
| 가점, 감점 항목 검출 | 추가 점수, 감점 요소를 검출하여 점수에 반영 합니다. |
| 타사 지원 내용  검출 | 자기소개서 내용 중에 언급되는 지원 회사와 실제로 지원한 회사가 다를 경우를 검출합니다. |
| 불성실 자기소개서검출 | 평가모델을 설정할 수 있는 메뉴를 제공 합니다. |
| 개발도구 | 훈련문장 자동 추출 도구(Tool) | 수집된 훈련 문장을 활용하여 평가기준에 부합하는 훈련 문장들을 자동 추출 합니다. |
| 문장단위 평가모델 생성 도구(Tool) | 반복적인 문장단위 평가모델 생성, 훈련(Training), 평가 작업을 자동화 합니다. |
| 문서단위 평가모델 생성 도구(Tool) | 반복적인 문서단위 평가모델 생성, 훈련(Training), 평가 작업을 자동화 합니다. |

[표 1] 주요 Feature

# **시스템 구성**

1.2절에서 언급했듯이 솔루션은 “어플리케이션”과 “평가엔진”으로 구성되어 있습니다. 전체 구성도는 [그림 4]와 같습니다.



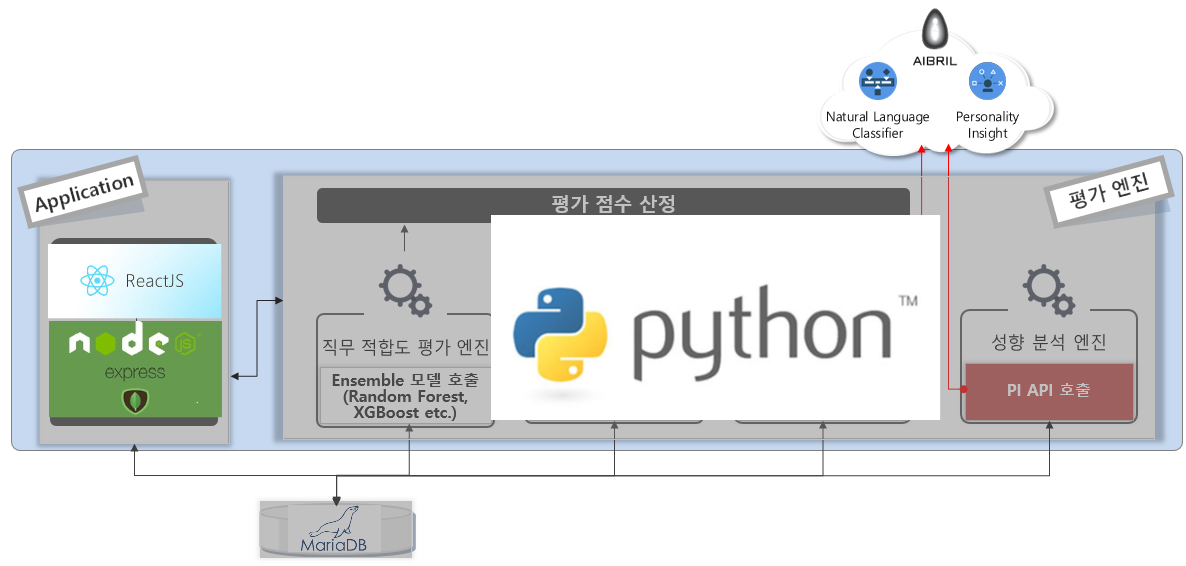
[그림 4] 시스템 구성도

평가엔진은 자기소개서 전체 내용을 기반으로 평가하는 “문서 평가 엔진”과 문장 단위로 평가를 하는 “문장 평가 엔진” 그리고 NCS데이터와 기업의 Job Description을 이용하여 직무 적합도를 평가하는 “직무 적합도 평가 엔진”으로 구성되어 있습니다.

“어플리케이션” 및 “평가엔진” 개발에 사용된 개발도구 및 데이터베이스는 [표 2]와 같습니다.

|  |  |
| --- | --- |
| **구성모듈** | **설명** |
| 어플리케이션 | Node.js, ReactJS |
| 평가엔진 | Python |
| DB | Maria DB |

[표 2] 개발도구, 데이터베이스



[그림 5]개발 플랫폼 및 언어

[그림 5]는 어플리케이션, 평가엔진 개발에 사용된 개발 언어 및 데이터베이스 정보를 기준으로 작성된 구성도 입니다.

2.1 ~ 2.5절을 통해 “어플리케이션”, “평가엔진”, “평가모델”, “점수화(scoring)”, “키워드 추출 및 요약문 생성”에 대한 설명을 제공합니다.

# 어플리케이션

어플리케이션은 사용자 UI를 구성하는 “Frontend”와 “Business Logic, 평가엔진 및 DB 연동” 기능이 구현되어 있는 Backend로 구성되어 있습니다.

Node.js로 개발된 어플리케이션에서 python으로 개발된 평가엔진의 호출은 “PythonShell”을 이용합니다 (“<https://github.com/extrabacon/python-shell>” 참고).

어플리케이션 소스 구조 및 Customization에 필요한 상세 설명은 “**3 어플리케이션 (Application)**"을 통해 제공 됩니다.

# 평가 엔진

평가 엔진은 “문장평가엔진”과 “문서평가엔진”으로 구성되어 있습니다.

# 평가모델 생성, 학습

문장단위 평가모델과 문서단위 평가모델을 생성, 학습 시키는 방안에 대한 기본적인 설명을 제공 합니다.

# 점수 산정(scoring)

문장평가엔진(w/문장 단위 평가모델), 문서평가엔진(w/문서 단위 평가모델), 직무적합도평가엔진을 통해 평가된 결과를 이용하여 최종 평가 점수를 산정 합니다.

# 키워드(Keywords), 요약문(Summarization)

# **어플리케이션 (Application)**

2.1절에서 설명했듯이 어플리케이션은 사용자 UI를 구성하는 “Frontend”와 “Business Logic, 평가엔진 및 DB 연동” 기능이 구현되어 있는 Backend로 구성되어 있습니다.

어플리케이션의 소스 구성 및 설명, 주요 Use Case에 대한 Sequence Diagram, Customization 영역 및 예제 코드를 제공하겠습니다.

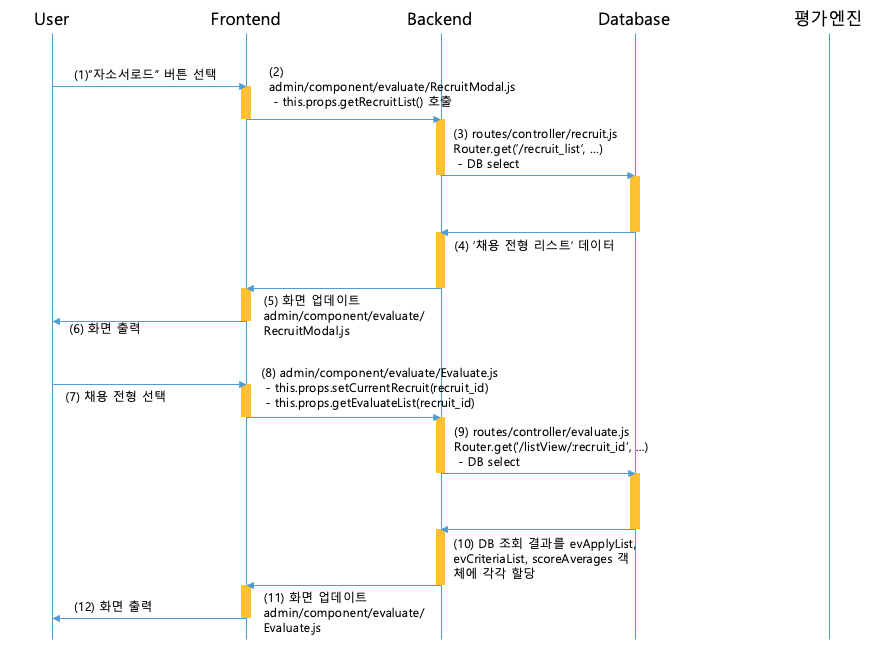
# 구성

어플리케이션은 Front-end는 React.js로 Back-end는 node.js(express)로 개발되어 있고, “python-shell”모듈을 이용해서 python으로 개발된 평가엔진과 연동합니다

# 주요 Sequence

데이터 로드(load), 평가 실행, 상세 화면 (평가결과조회, Word Cloud, Personality Insights)에 대한 Sequence Diagram을 제공합니다.

1. 평가데이터 로드

사용자가 데이터 로드 화면에 진입하여 평가데이터를 로드(load)하는 Sequence는 [그림 17]과 같습니다

[그림 17] 평가데이터 로드(load)

1. “자소서로드” 버튼을 선택합니다.
2. “자소서로드” 버튼을 클릭하면, admin/component/evaluate/RecruitModal.js 화면(모달)이 열리면서 reducer에 등록된 getRecruitList() 함수가 호출됩니다.

# 레퍼런스(Reference)

* + 1. 주요 소스 설명

어플리케이션을 수정 시 사용될 주요 source들에 대한 설명은 [표 5]를 참고해 주십시오.

|  |  |  |
| --- | --- | --- |
| Name | Parameter | Description, Example |
| client/src/App.js | Input: Component 파일  Output: index.html 화면 | import Login from "./components/auth/Login";   * Component 구성 화면 Import   <Route exact path="/login" component={Login} />   * Router할 Path 지정을 통해 화면 rendering   <Provider store={store}>   * Component에 Redux store를 매핑 |
| : | : | : |

[표 5] 주요 소스 설명

# **평가 엔진**

2.2절, 2.3절에서 설명했듯이 평가엔진은 “문장 단위 평가”, “문서 단위 평가”, “개인 성향 분석”을 수행 합니다. 본 장에서는 평가 엔진 소스 구성 및 설명, 주요 Use Case에 대한 Sequence Diagram, Customization 영역 및 예제 코드를 제공하겠습니다.

# 구성

평가엔진 패키지(package)를 구성하는 클래스(class)와 각 클래스의 기능은 [표 6]과 같습니다.

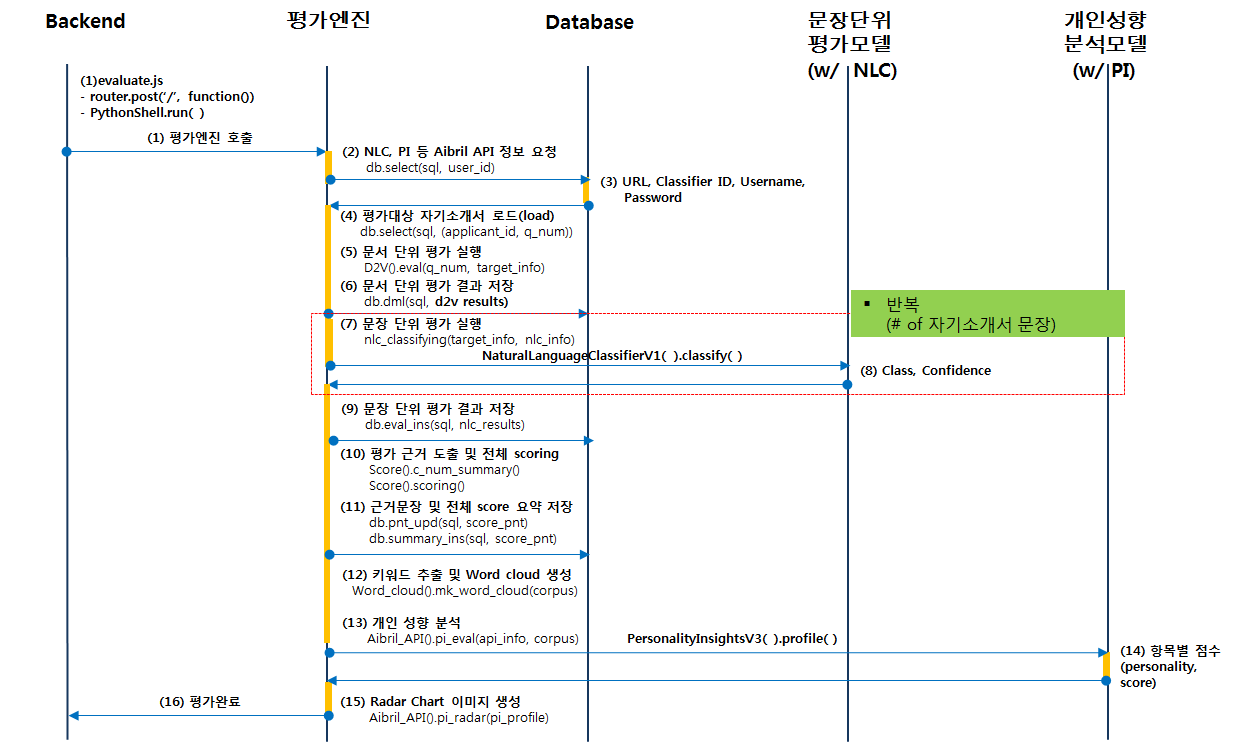
|  |  |  |
| --- | --- | --- |
| 이름 | 기능 | 설명 |
| Aibril\_API | 개인 성향 분석  문장 단위 평가 | * Watson Personality Insight API를 활용한 개인 성향 분석 * Watson Natural Language Classifier API를 활용한 문장 단위 평가 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

[표 7] 주요 구성요소, 기능

# 주요 Sequence

자기 소개서가 평가되는 Sequence에 대해서 설명합니다. 본 절에서는 생성되어 있는 “문장단위평가모델”과 “문서단위평가모델”을 사용하여 점수화하는 과정을 설명합니다.

“문장단위평가모델” 생성과 관련된 내용은 “8.**문장단위평가모델 개발 툴(tool)**”을 “문서단위평가모델” 생성과 관련된 내용은 “9.**문서단위평가모델 생성**”을 참고하시면 됩니다.



[그림 28] 자기소개서 평가

1. 평가엔진 호출
   * + Backend에서 “PythonShell.run( )”을 이용해서 평가엔진을 호출합니다.
     + 평가엔진 호출과 관련된 전반적인 흐름은 “3.2.2 평가 실행”을 참고하시면 됩니다.
2. , (3) NLC, PI 정보 요청
   * + PythonShell.run( )”을 호출할 때 제공되는 사용자ID(user\_id)를 이용하여 DB에 저장된 평가 항목별 NLC, PI 등 Aibril API Credential 정보를 load 합니다.

# 레퍼런스(Reference)

어플리케이션에서 평가엔진을 호출할 때 사용되는 API에 대한 설명과 예제 코드를 제공합니다.

|  |  |  |
| --- | --- | --- |
| Name | Parameter | Description, Example |
| Excel().Import(file\_name, Sheet) | Input: file\_name, Sheet  Output: file contents list | db\_info.xlsx 파일의 Sheet1 시트에서 db 정보를 import  Excel().Import("db\_info.xlsx", 'Sheet1') |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# **데이터베이스(DB)**

# 테이블정의서

DB 구성에 대해 설명 설명합니다. 테이블 정의서는 [표 7]과 같습니다.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 테이블명 | 컬럼명 | 컬럼타입 | Null허용 | Key | Default값 | Extra |
| sse\_apply\_info  (채용 지원자 정보) | apply\_dt | varchar(8) | NO |  | NULL |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

[표 8] 테이블 정의서

# **개발 환경 구성**

“Aibril HR for recruit”의 소스와 라이브러리를 활용하여 프로젝트를 수행할 때 필요한 개발 환경 구성에 대한 정보를 제공합니다.

# 시스템 사양

개발 환경 구성에 필요한 물리적 서버의 최소 사양은 [표 7]과 같습니다.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **구분** | **CPU** | **Memory** | **HDD** | **수량** | **SW** |
| 개발 서버 | 8 core \* 1 | 32 GB | 300 GB  이상 | 1 ea | Enterprise Linux or Ubuntu,  Node.js (10.\* - LTS),  Python (3.6.\*),  Maria DB (10.\*),  Redis 등 |

[표 7] 개발서버사양

Ubuntu를 사용할 경우에는 14.04 LTS/16.04 LTS 버전을, Enterprise Linux를 사용할 경우에는 RHEL 6(64-bit) 버전을 권장 합니다. 별도의 WEB 서버를 구성할 필요는 없습니다.

어플리케이션은 모던 자바스크립트(>=ES6) 동작을 보장하는 브라우저 기준으로 제작되었으며, 최신 버전의 크롬을 통해 접속하는 것을 권장합니다.

# 어플리케이션

어플리케이션은 Back-end와 Front-end 소스가 구분되어 있으며, 각각 Node.js와 ReactJS로 개발되어 있습니다. “6.2절”에서는 해당 프레임워크와 모듈(패키지)의 설치 방법에 대해 설명합니다.

1. Node.js 설치(Back-end)
   * + Node.js 공식 다운로드 페이지 : <https://nodejs.org/ko/download/>
     + Node.js는 10.\*의 LTS버전 설치를 권장합니다.
     + Enterprise Linux 사용시 공식 저장소의 Binary 파일을 사용한 Manually-install을 권장합니다 : <https://github.com/nodesource/distributions/blob/master/README.md>
     + yarn을 사용하여 모듈을 설치할 경우 <https://yarnpkg.com/lang/en/>을 참고하십시오.
     + 정상적인 설치 확인을 위해 CLI에서 다음 명령을 수행합니다 :

$ node --version && npm --version

1. React.js 설치(Front-end)
   * CLI에서 다음 명령을 수행합니다 :

$ npm install –g create-react-app react react-dom

1. 개발을 위한 디렉토리를 생성합니다.
2. 제공된 소스를 (3)에서 생성한 디렉토리에 압축 해제합니다.
3. Back-end의 모듈 설치
   * CLI에서 다음 명령을 수행합니다 :

$ cd {프로젝트 경로}/APPLICATION

$ npm install

* + VS-Build 관련 에러 발생 시 다음 명령을 수행합니다.

$ npm install -g --production windows-build-tools

1. Back-end 환경 설정
   * 어플리케이션에서 DB와 Python 환경에 접근하기 위해 해당 정보 및 경로를 설정해야 합니다.
   * APPLICATION/config/ 디렉토리의 keys\_dev.js 에는 개발 환경 정보를, keys\_prod.js에는 운영 환경 정보를 기입하면 됩니다.
   * APPLICATION/ENGINE/variables.xlsx에 DB 접속 정보를 수정합니다.
2. 클라이언트(client)용 패키지(package) 설치
   * CLI에서 다음 명령을 수행합니다 :

$ cd {프로젝트 경로}/APPLICATION/client

$ npm install

1. react-scripts의 빌드 설정 변경
   * Front-end 빌드는 react-scripts 모듈을 통해 컴파일 됩니다. 해당 값을 프로젝트에 맞게 수정하기 위해 CLI(혹은 IDE)에서 다음 과정을 수행합니다 :

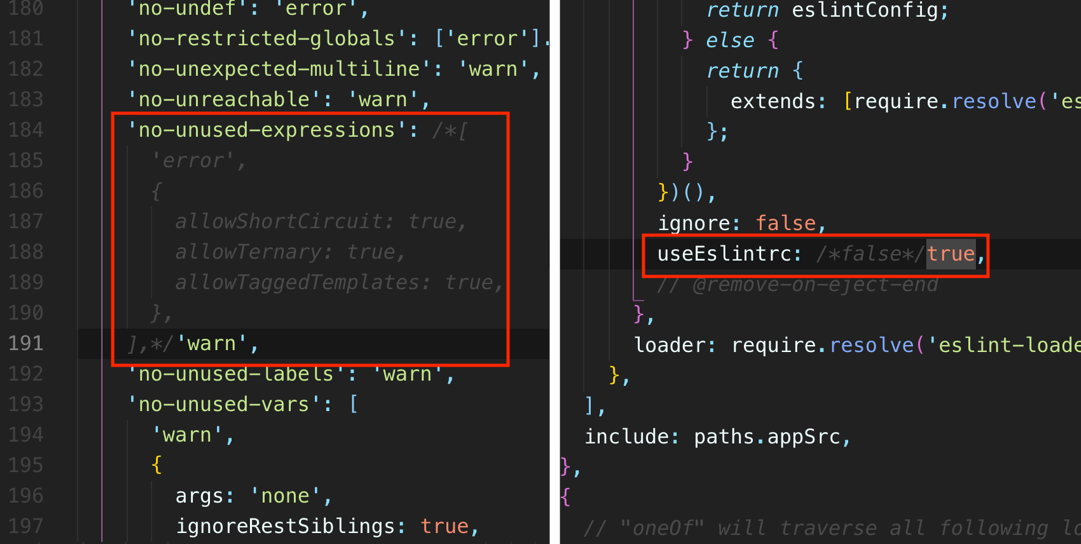
$ cd {프로젝트 경로}/APPLICATION/client/node-modules/

$ vi eslint-config-react-app/index.js

에디터에서 ‘no-unused-expressions’의 값을 ‘warn’으로 변경

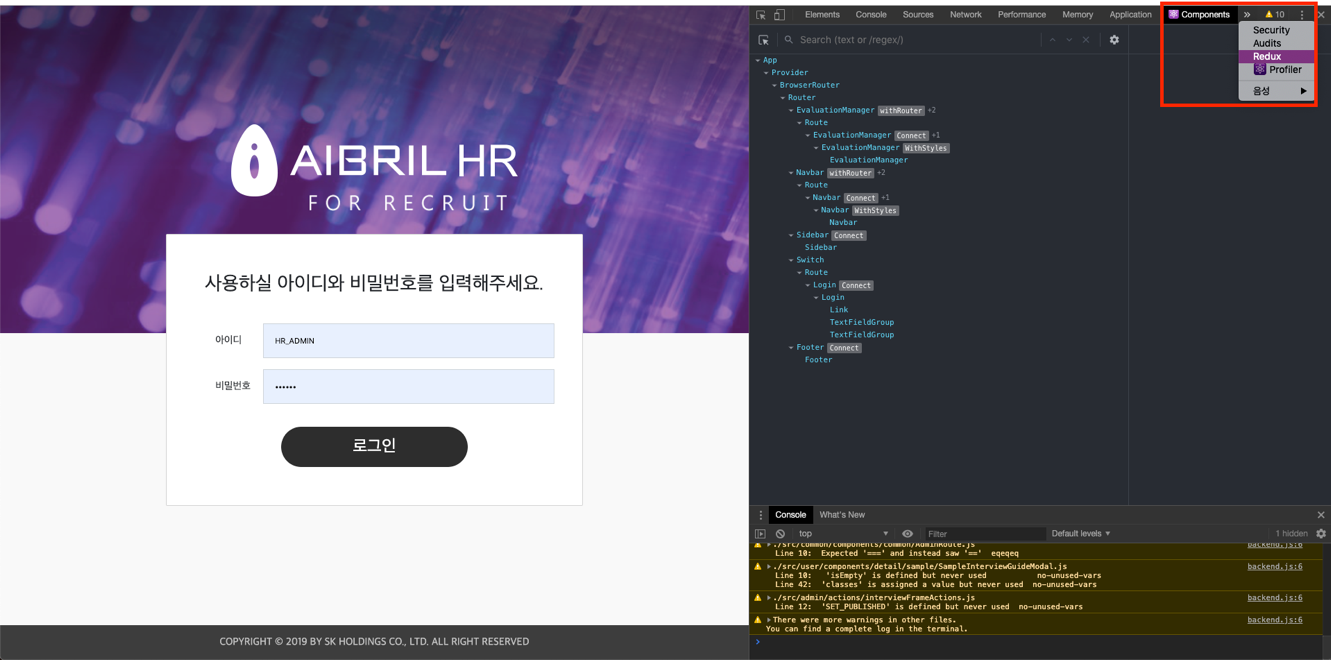
$ vi react-scripts/config/webpack.config.js

에디터에서 ‘useEslintrc’의 값을 true로 변경



[그림 6-1] react-scripts와 ES-lint 설정 값의 변경

1. 개발 도구(React/Redux) 설치
   * 최신 버전의 크롬에선 React 개발에 유용한 React/Redux 개발 도구를 지원합니다.
   * 해당 확장 기능은 크롬 웹스토어에서 ‘React Developer Tools’, ‘Redux DevTools’를 검색하여 설치합니다:
   * 설치가 완료되면 크롬 개발자 도구(F12)에서 [그림 6-2]와 같이 확인 가능합니다.



[그림 6-2] React/Redux 개발 도구의 설치 확인

# 평가 엔진

평가 엔진은 파이썬(Python)으로 개발되어 있습니다. “6.3”절에서는 평가 엔진 개발 환경 설정에 대해 설명합니다.

1. 파이썬 설치
   * Python 공식 다운로드 페이지 : <https://www.python.org/downloads/>
   * Python은 3.6.\* 버전 설치를 권장합니다.
2. Windows 환경에서 한국어 형태소 분석기 Konlpy 패키지 설치를 위한 환경 설정
   * JDK(Java Development Kit) 설치:

<https://www.oracle.com/technetwork/java/javase/downloads/index.html>

* + JAVA\_HOME Path 설정 방법: <https://tocomo.tistory.com/3>
  + Jpype1 설치

설치 파일 링크: <https://www.lfd.uci.edu/~gohlke/pythonlibs/#jpype>

설치 방법 링크: <https://ellun.tistory.com/46>

1. 패키지 설치
   * 파이썬 패키지 설치 시 pip-freeze 기능을 이용한 자동 설치를 권장합니다. 파이썬 가상 환경이 있다면 활성화 한 후, CLI에서 다음 명령을 수행합니다 :

$ pip install -r {프로젝트 경로}/APPLICATION/requirements.txt

* + GUI에서 수동 설치가 필요하다면 PyCharm IDE를 통해 설치하는 것을 권장합니다. 필수 패키지들의 목록 및 버전은 모두 {프로젝트 경로}/APPLICATION/requirements.txt 파일에 기재되어 있습니다.

# 데이터베이스(Database) 설치

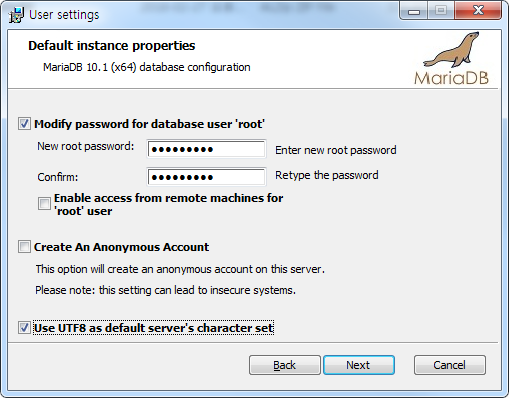
Maria DB를 기준으로 DBMS 설치 및 초기 설정을 설명합니다.

1. Maria DB 공식 다운로드 페이지 : [https://downloads.mariadb.org](https://downloads.mariadb.org/)
2. 설치
   1. 각 운영체제에 알맞는 설치 방법은 다음 링크를 참고하여 진행합니다 :

<https://mariadb.com/kb/en/library/binary-packages/>

* 1. 다음은 GUI(Windows) 환경에서 설치 시 설정해야 할 사항들입니다..
     1. Character Set 설정

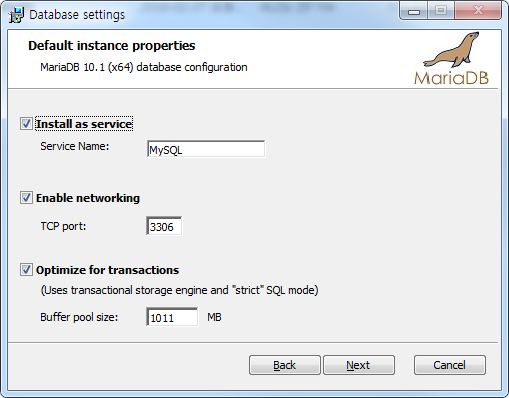
“Use UTF8 as default server’s character set”을 체크 합니다.



[그림 6-3] Character Set 설정

* + 1. Service Name 설정

“Service Name”에 윈도우 서비스로 등록될 서비스 명 입력



[그림 6-4] Service Name 설정

1. User 및 Database 생성

ROOT 권한으로 Maria DB에 접속하고, 사용자 및 DB를 생성하는 과정을 설명합니다.

Windows 이외의 환경은 다음 링크를 참고하십시오 :   
<https://mariadb.com/kb/en/library/documentation/mariadb-administration/getting-installing-and-upgrading-mariadb/>

1. my.ini 파일 설정
   * + {DB 설치 디렉토리}/data/my.ini 파일을 열어 [그림 6-8]의 내용을 추가합니다. (또는 시작 🡪 모든 프로그램 🡪 MariaDB 10.\* 🡪 my.ini 바로가기 선택)

[client]

default-character-set=utf8mb4

[mysql]

default-character-set=utf8mb4

[mysqld]

collation-server=utf8mb4\_unicode\_ci

init-connect='SET NAMES utf8mb4'

character-set-server=utf8mb4

lower\_case\_table\_names=1

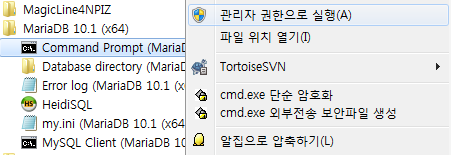
innodb\_file\_format=barracuda

innodb\_file\_format\_max=barracuda

innodb\_large\_prefix=on

[그림 6-5] ini 파일 설정

1. MariaDB 서비스 재시작
   * 1. my.ini에 추가된 내용을 적용시키기 위해 서비스를 재시작 해야합니다.
     2. “시작” 🡪 “프로그램 및 파일 검색” 창에 “services.msc” 입력 후 엔터
     3. “6.3”의 5번째 설치화면에서 설정한 윈도우 서비스명(기본 MySQL)을 찾아서 재실행
2. MariaDB 관리자 권한으로 접속
   * 1. “시작” 🡪 “모든 프로그램”🡪 MariaDB10.\* 🡪 “Command Prompt”를 마우스 우클릭 🡪 관리자 권한으로 실행



[그림 6-6] 관리자 권한 접속

* + 1. CLI에서 다음 명령을 수행합니다 :

$ mysql –u root -p {ROOT PASSWORD}

* + 1. 사용자(hr) 생성 및 패스워드 생성

create user 'hr' IDENTIFIED BY ‘{PASSWORD FOR hr}’;

* + 1. DB 설정 확인

SHOW VARIABLES WHERE Variable\_name LIKE 'character\\_set\\_%' OR Variable\_name 'collation%';

결과가 [그림 6-10]과 동일하면 정상적으로 설정된 것 입니다.

+---------------------------- +--------------------- +

| Variable\_name | Value |

+---------------------------- +--------------------- +

| character\_set\_client | utf8mb4 |

| character\_set\_connection | utf8mb4 |

| character\_set\_database | utf8mb4 |

| character\_set\_filesystem | binary |

| character\_set\_results | utf8mb4 |

| character\_set\_server | utf8mb4 |

| character\_set\_system | utf8 |

| collation\_connection | utf8mb4\_general\_ci |

| collation\_database | utf8mb4\_unicode\_ci |

| collation\_server | utf8mb4\_unicode\_ci |

+---------------------------- +--------------------- +

[그림 6-7] DB 설정 확인

* + 1. DB 생성

UTF-8 확장 문자열을 저장할 수 있는 포맷으로 지정해서 생성합니다.

set global innodb\_file\_format = BARRACUDA;

set global innodb\_file\_format\_max = BARRACUDA;

set global innodb\_large\_prefix = ON;

create database hr

DEFAULT CHARACTER SET utf8mb4

DEFAULT COLLATE utf8mb4\_bin;

* + 1. 사용자 “hr”에게 생성된 DB의 권한 부여

GRANT ALL ON hr.\* to 'hr';

* + 1. 생성된 DB 접속 확인  
       “exit” 명령어를 입력하여 Console을 빠져 나온 뒤, 사용자 “hr”로 정상 접속되고 생성된 “hr” DB가 사용 가능한 지 확인해 봅니다.

mysql -u hr -p {PASSWORD FOR hr}

use hr;

1. HeidiSQL 설치 및 데이터 불러오기

생성된 DB를 관리할 수 있는 HeidiSQL 설치, 설정, 제공된 sql 파일을 DB에 로딩(loading)하는 방법에 대해 설명합니다.

1. HediSQL 다운로드 <https://www.heidisql.com/download.php>
2. HeidiSQL 설정 <https://blog.naver.com/knh_0313/221147690147>
3. HeidiSQL에서 [파일]-[SQL 파일 불러오기]로 함께 제공된 hr\_recruit\_db.sql 파일 실행

# 어플리케이션 실행

“6.1절 ~ 6.4절”의 내용에 따라 어플리케이션 실행에 필요한 구동 환경이 갖추어 지면, CLI에서 스크립트 명령을 입력하여 어플리케이션을 실행할 수 있습니다.

1. 개발 버전 어플리케이션 실행
   * + CLI에서 다음 명령을 수행합니다 :

$ cd {프로젝트 경로}/APPLICATION

$ npm run dev

* + - 빌드가 완료되면, <http://localhost:3000> 에 접속하여 실행을 확인합니다.

1. 프로덕션 버전 어플리케이션 실행
   * + Windows 환경이라면 다음 과정을 수행합니다 :

$ vi {프로젝트 경로}/APPLICATION/package.json

prod-server의 값을 ‘set NODE\_ENV=production && node server.js’로 수정

* + - CLI에서 다음 명령을 수행합니다 :

$ cd {프로젝트 경로}/APPLICATION

$ npm run prod

* + - 빌드가 완료되면, <http://localhost:5000> 에 접속하여 실행을 확인합니다.



[그림 6-7] 어플리케이션 실행 화면