|  |
| --- |
|  |
| oopscraft-application |
| Application Architecture Definition |
|  |
|  |
|  |

|  |
| --- |
|  |

1. Architecture Overview (아키텍처 개요) 2

1.1. Standalone Application 2

1.2. Embedded Web Server 2

1.3. Spring Framework 2

1.4. Spring Security 3

1.5. JPA(with spring-data) and Mybatis 3

1.6. Commonly Used Component Modules 3

2. Application Structure (어플리케이션 구조) 3

2.1. Directory and File Structure (디렉터리 및 파일 구조) 3

2.2. Java Package Structure (JAVA 패키지 구조) 3

3. Application Configuration (어플리케이션 설정) 4

3.1. Configuration Value 4

3.2. Embedded WebServer (임베디드 웹서버 설정) 4

3.3. DataSource 설정 4

3.4. JPA Entity Manager 설정 4

3.5. MYBATIS Session Factory 설정 4

3.6. Message Source의 설정 4

4. Application Build 5

4.1. Local Environment 5

4.2. Server Environment 5

5. Application Start and Stop 5

5.1. Start Application 5

5.2. Check Application Status 5

5.3. Stop Application 5

5.4. View Application Log 5

6. Check Web Application Channel 6

6.1. Admin Channel 6

6.2. Service Channel 6

7. Extended Development (확장 구현) 6

# Architecture Overview (아키텍처 개요)

배치실행 어플리케이션이란 개발된 배치프로그램의 스케줄링 및 실행을 하기 위한 Java Standalone 어플리케이션으로(웹과 비교하자면 배치실행 어플리케이션은 웹서버, 각 배치프로그램은 각각의 웹화면에 비교할수 있음) 웹 기반 배치프로그램 실행 및 관리를 위한 Tomcat 웹서버 및 배치프로그램에서의 데이터베이스 연동을 위한 Mybatis가 Embedded된 형태의 독립 어플리케이션이다. 대략적인 내부 구조는 다음과 같다.



## Standalone Application

웹서버 또는 WAS 없이 독립적으로 실행되는 Standalone 어플리케이션 개발을 위한 플랫폼이다.

## Embedded Web Server

Embedded Webserver 인 tomcat을 이용한 웹 어플리게이션을 구한다.

## **S**pring Framework

웹 어플리케이션은 엔터프라이즈 환경에서 적합한 spring-framework를 표준으로 구성한다.

## Spring Security

웹 어플리케이션 및 오픈 API의 보안 구현은 spring security 라이브러리 기반으로 구현된다.

## JPA(with spring-data) and Mybatis

## Commonly Used Component Modules

# **Application Structure (어플리케이션 구조)**

## Directory and File Structure (디렉터리 및 파일 구조)

|  |  |
| --- | --- |
| **디렉토리** | **설명** |
| ./conf | 설정파일 디렉토리 |
| ./conf/application.xml | 어플리케이션 설정파일 |
| ./conf/application.properties | 설정파일(port,데이터베이스접속정보 등) |
| ./conf/mybatis.xml | MYBATIS 설정파일 |
| ./webapp/application/\* | Embedded Tomcat에서 제공되는 웹어플리케이션(관리화면,RESTful API) |
| ./application.sh | 어플리케이션 구동 쉘스크립트 |
| ./chamomile-batch-${version}.jar | 어플리케이션 JAR 바이너리 파일(배치프로그램 개발시에도 해당 libray 사용) |
| ./build.sh | 빌드 자동화 스크립트(개발환경에서만 사용) |
| ./run-batch.sh | 배치Workflow를 Command로 실행하기 위한 쉘스크립트 |
| **./depoly** | **개발된 배치프로그램의 jar가 deploy되는 디렉토리** |
|  |  |

## Java Package Structure (JAVA 패키지 구조)

Deploy된 디렉터리 및 파일 구조는 아래와 같다.

|  |  |
| --- | --- |
| **디렉토리** | **설명** |
| ./conf | 설정파일 디렉토리 |
| ./conf/application.xml | 어플리케이션 설정파일 |
| ./conf/application.properties | 설정파일(port,데이터베이스접속정보 등) |
| ./conf/mybatis.xml | MYBATIS 설정파일 |
| ./webapp/application/\* | Embedded Tomcat에서 제공되는 웹어플리케이션(관리화면,RESTful API) |
| ./application.sh | 어플리케이션 구동 쉘스크립트 |
| ./chamomile-batch-${version}.jar | 어플리케이션 JAR 바이너리 파일(배치프로그램 개발시에도 해당 libray 사용) |
| ./build.sh | 빌드 자동화 스크립트(개발환경에서만 사용) |
| ./run-batch.sh | 배치Workflow를 Command로 실행하기 위한 쉘스크립트 |
| **./depoly** | **개발된 배치프로그램의 jar가 deploy되는 디렉토리** |
|  |  |

# Application Configuration (어플리케이션 설정)

## Configuration Value

어플리케이션에서 사용할 설정 값들을 설정한다.

## Embedded WebServer (임베디드 웹서버 설정)

## DataSource 설정

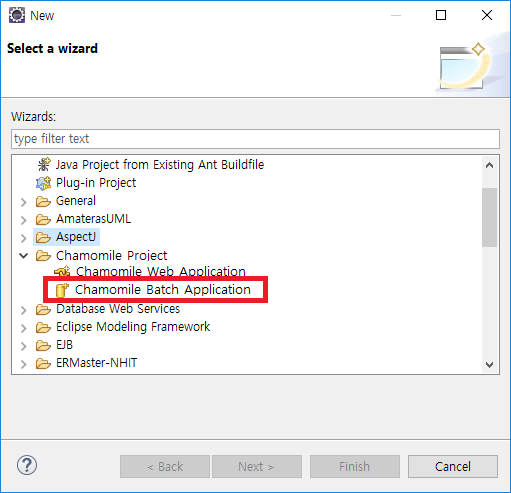
## JPA Entity Manager 설정

## MYBATIS Session Factory 설정

## Message Source의 설정

# Application Build

## Local Environment



## Server Environment

# Application Start and Stop

## Start Application

user@localhost> ./application.sh start

## Check Application Status

user@localhost> ./application.sh status

## Stop Application

user@localhost> ./application.sh stop

## View Application Log

user@localhost> ./application.sh log

# Check Web Application Channel

## Admin Channel

## Service Channel

# Extended Development (확장 구현)

개발가이드를 참조한다.