**IB 개발 표준 정의서**

**(Coding Standards)**

**프로젝트** IB 개발표준정의서

**작성자** 박지철, 이용혁

**작성일** 2012-02-16

**작성부서** 서비스개발실

**문서버전** Ver.1.0

변경내용

Ex) 2012-02-16 / 문서 작성 사유 기입 / 홍길동

1. 2012-02-16 / 문서 생성 / 박지철

**목차**

1. **개발 표준 정의서**
   1. **서문**
   2. **저작권**
   3. **문서 이력 관리**
   4. **문서의 목적**
   5. **팀 표준을 만드는 방법**
2. **Platform**
   1. **운영 환경**
   2. **개발 환경**
3. **표준**
   1. **서블릿(servlet)**
   2. **웹 표준 준수**
4. **소스 코드 관리**
   1. **Revision Control**
5. **산출물**
   1. **소스 코드**
   2. **설치 파일**
   3. **스크립트**
   4. **문서**
6. **개발 지원 서버**
   1. **SVN 서버**
   2. **데이터베이스 서버**
7. **구조**
   1. **시스템 구성**
   2. **소프트웨어 구조**
   3. **패키지 구조**
   4. **프로그램 구조**
8. **인코딩**
   1. **텍스트 파일 인코딩**
   2. **HTTP Request/Response 인코딩**
9. **Coding Convention**
   1. **공통**
      1. **공백(Spaces and Spacing)**
      2. **들여쓰기, 중괄호, 정렬(Indentation, Braces, and Alignment)**
   2. **Naming Convention**
      1. **Action Name**
      2. **JSP Name**
      3. **Class, Interface Name**
      4. **Method Name**
      5. **변수(Variable), 인자(Argument)**
      6. **Constant Name**
      7. **Package Name**
      8. **Domain Class and Field Name**
   3. **Database**
   4. **JavaDoc 주석**
   5. **HTML Coding**
   6. **JAVA Naming**
      1. **Class, Interface File**
      2. **JSP File**
      3. **Model File**
      4. **Procedure File(Mapper)**
      5. **JavaScript File**
      6. **Image File**
10. **개발 표준 정의서**
    1. **서문**  
       이 문서는 itemBay 서비스개발팀에 의해 작성되었으며 문서에서 다루는 대부분의 정보는  
       서비스개발팀의 경험과 자바 표준 가이드 라인과 다른 여러 가지 문서 및 소스를  
       기초로 하여 작성되었습니다.
    2. **저작권**  
       저작권 정보를 유지한다면 이 문서를 비상업적인 용도로 어디든지 사용할 수 있습니다.  
       이 문서는 현 시점을 기준으로 작성되었으며, 이 문서를 참고 함으로써 발생하는   
       어떠한 손실에도 저자는 책임이 없습니다.
    3. **문서 이력 관리**  
       문서를 수정하였을 경우 이력 양식에 기본 수정 정보를 기입하여야 합니다.  
       예) 2010-07-30 / 문서 작성 사유 기입 / 홍길동
    4. **문서의 목적**  
       개발자들이 알아야 할 사항들을 정리하고, 통일성 있는 코드를 생산해 내기 위한 최소한의   
       규칙을 정하고 유지보수에 좋고 신뢰받는 코드를 개발하기 위한 문서입니다.
    5. **팀 표준을 만드는 방법**반드시 모두가 표준안에 동의를 한 상태여야 하며 표준안이 마련되면   
       모두에게 전달해야 합니다.
11. **Platform**
    1. **운영 환경**- CentOS  
       - MS-SQL 2005 Server(추후 2008 버전으로 업그레이드 예정)  
       - MongoDB(NoSQL 기반)
    2. **개발 환경**- JDK 1.6  
       - JBoss Application Server 5.1  
       - Spring 3 – 객체 관리, 트랜잭션 관리, Form validation  
       - MyBATIS 3 – SQL문 관리(MS-SQL 쿼리 관리)  
       - JSTL 1.1 – JSP에서 유용한 태그들 제공  
       - jQuery 1.4 – AJAX와 HTML 객체들을 동적으로 관리
12. **표준**
    1. **서블릿(servlet)**- Servlet 2.4/JSP 2.0을 사용한다.
    2. **웹 표준 준수**  
       XHTML 1.0와 CSS2를 사용하여 컨텐츠와 스타일을 분리하며,   
       다음 브라우저에서의 동작을 테스트해야 한다.  
         
       - Internet Explorer (6 이상)  
       - Firefox  
       - Chrome  
       - Opera
13. **소스 코드 관리**
    1. **Revision Control**- Subversion 사용  
       - 저장소 :
14. **산출물**
    1. **소스 코드**
    2. **설치 파일**- 사용되는 제품들(JDK, Apache, JBoss, MongoDB, …)의 설치 파일  
       - Web Archive(war) 파일
    3. **스크립트**- 애플리케이션 설치 스크립트  
       - 백업 스크립트 (데이터베이스, 첨부파일, …)  
       - 복원 스크립트
    4. **문서**HTML 또는 MS Word를 사용하여 작성해야 한다.  
         
       - 아이템베이 개발계획서  
       - 아이템베이 개발표준 정의서  
       - 아이템베이 사용자 가이드  
       - 아이템베이 관리자 가이드  
       - 아이템베이 개발자 가이드  
       - 아이템베이 인터페이스 정의서
15. **개발 지원 서버**
    1. **SVN 서버**
    2. **데이터베이스 서버**
16. **구조**
    1. **시스템 구성**
    2. **소프트웨어 구조**
    3. **패키지 구조**패키지 이름은 com.itembay 로 시작한다.  
         
       - com.itembay.web : 사용자 Web Application에서 사용  
       - com.itembay.web.common : 사용자에서 사용되는 공통 패키지  
       - com.itembay.admin : 관리자 Web Application에서 사용  
       - com.itembay.admin.common : 관리자에서 사용되는 공통 패키지
    4. **프로그램 구조**하나의 프로그램은 다음과 같이 여러 파일로 구성된다.

|  |  |
| --- | --- |
| 파일 | 역할 |
| XXXController.java | 브라우저와의 데이터 교환, Request/Response/Session 처리 JSP와의 연계 |
| XXXService.java | 비즈니스 로직 인터페이스 |
| XXXServiceImpl.java | 실제 비즈니스 로직을 담당, Transaction 처리 |
| XXXDao.java | 데이터의 보관/조회를 담당, SQL문을 실행, 파일 시스템 접근 |
| sql-map.xml | SQL문과 캐싱 정책 등이 기술됨 |

1. **인코딩**
   1. **텍스트 파일 인코딩**모든 텍스트 파일(.java, .xml, .jsp, .html, .css, .js, …)의 인코딩은 UTF-8을 사용한다.
   2. **HTTP Request/Response 인코딩**  
      모든 요청/응답의 인코딩은 UTF-8을 사용한다.
2. **Coding Convention**
   1. **공통**
      1. **공백(Spaces and Spacing)**1) 코드 단락마다 공백 한 줄을 넣는다.  
           
          public / protected / private 한정자 사이를 공백 한 줄로 띄운다.  
           
         2) 공백 라인을 두 줄 이상 사용해서 트리 구조를 무너지게 하지 않는다.  
           
          private string Name;  
          protected string Name;  
          public string Name;  
           
         3) 키워드나 함수 이름과 그 다음에 나오는 시작 괄호 사이에 공백을 넣지 않는다.  
           
          private void HelloWorld()  
           
         4) 함수의 파라미터가 없을 때는 좌 괄호와 우 괄호 사이에 공백을 넣지 말고  
          파라미터가 여러 개일 경우 “,” 뒤에 공백을 삽입한다.  
           
          private void HellWorld()  
          private void HellWorld(string Name, DateTime NowDate)   
           
         5) 산술 연산자는 좌 우로 공백 하나를 사용한다.  
           
          int i = i + 1;
      2. **들여쓰기, 중괄호, 정렬(Indentation, Braces, and Alignment)**1) 들여쓰기는 탭 문자(4자리 공백)를 사용한다.  
           
         2) Scope의 레벨이 내려갈 때 마다 들여쓰기를 한다.  
           
         3) Case 문도 들여쓰기를 한다.  
           
         4) 시작 중괄호는 함수명과 같은 레벨에 선언한다.  
           
         private void HelloWorld() {  
          System.out.println("HelloWorld ");  
         }  
           
         if (true) {  
          return 1;  
         } else {  
          return 0;  
         }  
           
         5) 간단한 인라인 함수나 반복문은 중괄호를 같은 줄에 사용한다.  
           
         public string GetName() { return Name; };  
           
         6) 중괄호를 사용하지 않아도 되는 경우에도 사용한다.  
           
         while (true) {  
          return false;  
         }  
           
         7) 한 줄이 너무 길어 다음 줄에 쓰는 경우 수직정렬을 한다.  
           
         private void HellWorld(string Name,  
          DateTime NowDate)  
         }
   2. **Naming Convention**  
      **파스칼 케이싱(Pascal Casing)**  
       - 단어의 첫 번째 글자는 모두 대문자, 나머지는 모두 소문자  
       ex) BackColor  
        
      **캐멀 케이싱(Camel Casing)**  
       - 맨 첫 번째 단어를 제외한 단어의 첫 번째 글자는 대문자, 나머지는 모두 소문자  
       ex) backColor  
      1. **Action Name**  
         캐멀 케이싱(Camel Casing)을 사용한다.  
         ex) memberLogin, sellDetail, transactionDetail
      2. **JSP Name**캐멀 케이싱(Camel Casing)을 사용한다.  
         ex) memberLogin.jsp, sellDetail.jsp, transactionDetail.jsp
      3. **Class, Interface Name**  
         파스칼 케이싱(Pascal Casing)을 사용한다.  
         ex) public class HelloWorld, public interface HelloWorld
      4. **Method Name**  
         캐멀 케이싱(Camel Casing)을 사용한다.

|  |  |  |
| --- | --- | --- |
| Prefix | Explain | Method Name |
| Is | Boolean Return | isMember(string memberID) |
| Get | Value Return | getMemberName(string memberID) |
| Set | Value Setting | setMemberName(string memberName) |
| Select | Select Data | selectUserName(string memberID) |
| Insert | Insert Data | insertUser (string memberID) |
| Update | Update Data | updateUser (string memberID) |
| Delete | Delete Data | deleteUser (string memberID) |

* + 1. **변수(Variable), 인자(Argument)**  
       캐멀 케이싱(Camel Casing)을 사용한다.  
         
       1) 변수 선언은 한 줄에 하나씩 한 줄에 2개 이상 선언하지 않는다.  
        좋은 예 : private string memberID;  
        나쁜 예 : private string memberID, memberName;  
         
       2) 변수에 의미를 부여하며 축약하지 않으며 오탈 자에 주의한다.  
        좋은 예 : string address, string name   
        나쁜 예 : string addr, string nm  
         
       3) i, j, k 와 같은 한 글자로 이루어진 변수를 사용하지 않는다.  
        (의미가 필요 없는 반복 구문에서의 사용은 예외로 합니다.)  
        ex) for (int i = 0; i < count; i++)  
         
       4) 지역변수에 언더스코어( \_ ) 문자를 사용하지 않는다.  
        ex) private bool isLogin;  
         
       5) 변수 명을 예약 어와 유사하게 사용하지 않는다.  
        ex) private interface Interface;
    2. **Constant Name**  
       모두 대문자를 사용하며 단어 사이는 \_로 구분한다.  
       ex) public final int MAX\_APP\_SIZE = 12;
    3. **Package Name**  
       모두 소문자를 사용하며 영문자만 사용한다.  
       ex) com.itembay.web.common
    4. **Domain Class and Field Name**  
       가능한 한 연관되는 테이블 이름, 컬럼 이름과 동일하게 사용한다.  
       ex) vcMemberID, iPrice
  1. **Database**

|  |  |  |
| --- | --- | --- |
| Data Type | Prefix | Example |
| Tinyint | ti | tiSeq |
| Smallint | si | siSeq |
| Int | i | iSeq |
| Bigint | bi | biSeq |
| Char | c | cCode |
| Varchar | vc | vcTitle |
| Text | t | tContents |
| Bool | b | bFlag |

* 1. **JavaDoc 주석**모든 클래스, 인터페이스, 메소드에는 javadoc 주석을 반드시 작성한다.

|  |
| --- |
| /\*\*  \* <pre>  \* 1. MethodName : getItemBuyInfo  \* 2. Comment : 구매 물품 수정  \* 3. 작성일 : 2012. 02. 05  \* </pre>  \* @author jcpark  \* @param paramMap  \* @return Map  \* @throws Exception  \*/ |

* 1. **HTML Coding**  
     1) XHTML 1.0 Transitional 스펙을 준수한다.  
     2) html 태그에는 xmlns와 lang을 반드시 선언한다.  
     3) 모든 태그 이름에는 소문자를 사용한다.  
     4) 태그의 속성값을 지정할 때는 반드시 쌍따옴표를 사용한다.  
     5) 속성에는 반드시 값을 지정한다. 속성 이름만 지정할 수 없다.  
     6) 모든 태그에는 닫는 태그가 존재해야 한다.  
     7) 모든 태그는 포함 관계가 명확해야 한다. 두 태그의 시작과 끝이 중첩되면 안된다.  
     8) 디자인을 위한 태그나 속성은 사용하지 않는다.  
     9) HTML 문서에서는 최소화된 구조로 디자인을 제외한 문서의 내용만을 표시해야 한다.

1. **JAVA Naming**
   1. **Class, Interface File**1) Class Pages - HelloWorld 클래스는 HelloWorld.java파일로 저장한다.  
      2) Interface Pages - HelloWorld 인터페이스는 HelloWorld.java파일로 저장한다.
   2. **JSP File**1) Path : /WEB-INF/views/  
      2) JSP Pages Naming : PageName + 접미사 - 캐멀 케이싱(Camel Casing)  
      2) 접미사 구분

|  |  |  |
| --- | --- | --- |
| Type | Desc | ex |
| View | 일반 뷰 페이지 | PageName + View = mybayMainView.jsp |
| List | 리스트 페이지 | PageName + List = itemSellList.jsp |
| Detail | 리스트 상세 페이지 | PageName + Detail = itemSellDetail.jsp |
| Register | 등록 페이지 | PageName + Register = itemSellRegister.jsp |
| Edit | 수정 페이지 | PageName + Edit = itemSellEdit.jsp |
| Remove | 삭제 페이지 | PageName + Remove = itemSellRemove.jsp |

* 1. **Model File**  
     1) Path  
      - /model/emb/  
      - /model/entity/  
      - /model/mapper/  
       
     2) Mapper File Naming  
      파스칼 케이싱(Pascal Casing)을 사용한다.  
      ex) Help.xml  
      ItemBuy.xml  
      ItemSell.xml  
      Transaction.xml
  2. **Procedure File(Mapper)**  
     1) 기존 SP명 그대로 사용  
      - UP\_MYBAY\_ARRWAITAPPROVING -> up\_Mybay\_ArrWaitApproving 으로 변경  
       
     2) Mapper XML Tag Naming  
      - <insert> : insertItemSell  
      - <select> : selectItemSell  
      - <update> : updateItemSell  
      - <delete> : deleteItemSell
  3. **JavaScript File**  
     1) Path  
      - /resources/js/ : user file  
      - /resources/js/common : common file  
      - /resources/js/plugin : jquery plugin file  
      - /resources/js/json : json file  
       
     2) File Name  
      - /resources/js/ 루트에 생성한다.  
      ex) item.buyDetail.js  
      item.sellDetail.js  
      help.claimTradeRegister.js  
      transaction.forGiveTake.js  
       
     3) File Import  
      - language="javascript" - 삭제  
      ex) <script type="text/javascript" src="/resources/js/Jss/Plugins/jquery.js"></script>
  4. **Image File**  
     1) Path : /resources/image/

1. **JAVA Coding Convention**
   1. **Controller**
   2. **Service**
   3. **Service Implement**