**UB**

since June 2015

Engine, 강지현

본 문서에 가칭 UB 프로그램 개발에 관하여 명세한다.

문서의 개괄적 작성전에 요구사항을 우선 기입한다.

**Index**

현재 생략

**1. Introduction**

**Purpose**

**Definition**

**Glossary**

**References**

**2. Overall description**

**3. Specific requirements**

**External interface requirements**

**Functional requirements**

**Performance requirements**

**Design constraints**

**Other requirements**

<본문시작>

**1. Introduction**

서론 언급 현재 생략한다.

**1.1 Purpose**

현재 서술 생략

**1.2 Definition**

현재 서술 생략

**1.3 Glossary**

localhost: 도메인 혹은 root가 되는 주소이다.

root: 웹 서버 최상위 폴더

UB: 본 프로젝트의 가명, 웹 사이트인 동시에 웹 프로그램을 지칭하는 용어이다.

singleton: Singleton Form. 객체 하나만 생성. 추후 상세서술 예정

현재 서술 생략

**1.4 References**

현재 서술 생략

**2. Overall description**

현재 서술 생략한다

**3. Specific requirements**

**External interface requirements**

현재 서술 생략

**Functional requirements**

**Object Naming Rules**

**URL Contract**

URL 은 보통 다음과 같은 형태를 띈다.

domain/segment1/segment2/...

UB는 직관성과 그 밖의 이유 (후에 서술)에 의해 다음과 같이 URL을 사용한다.

domain/controller/parameter1/parameter2/...

segment1자리는 controller의 이름, 그 이후 모두 일종의 parameter취급을 한다.해당 요소 설명 시 자세히 설명한다.

**MVC Framework**

본 웹 프로그램은 Model View Structure 의 구조를 취하고 있다. MVC구조 전체를 관장하는 Core 객체로부터 기능이 실행된다. 각 요소를 다룰 때 더 상세히 언급한다.

**Gateway (root/index.php)**

localhost 가 호출되면 가장 먼저 root 폴더의 index.php 를 실행한다. 즉, root/index.php 가 UB의 Gateway 기능을 한다. 중요한 것은, 새로운 URL 로의 이동이 있을 시 (예: Anchor을 통한), 다시 가장먼저 Gateway를 실행한 후 요청에 맞게 프로그램을 실행한다는 것이다. Script를 이용한 페이지 요소 갱신 제외, 모든 링크 이동은 반드시 Gateway를 요청한다.

Gateway에는 다음과 같은 요소를 포함한다.

기능

Set Path: define함수로 프로그램에 쓰일 Path를 지정.

Core 객체 읽기

Core 인스턴스 생성 및 route 함수 실행

**Core**

Core는 UB의 최상위 Managing Object 이다.

Core는 다음과 같은 요소를 포함한다.

변수

static instance [$class]: instance배열이다. 클래스 이름을 selector로 가지고, UB에서 생성되는 모든 객체를 값으로 가진다. Static으로 구현하여 singleton 을 유지한다.

함수

getInstance($class): 객체 인스턴스를 생성하는 함수. load 함수를 포함한다.

load($class):생성할 인스턴스에 관한 정보를 읽는 역할을 하는 함수이다. require\_once 함수 활용

기능

Config와 Router객체를 생성하여 정보를 얻고, Router객체의 route함수를 실행한다.

**Config**

UB 실행에 관한 여러 Configuration 정보를 담고있을 객체이다.

app폴더의 config.php 를 참조한다.

추후 개발 요소로 남겨둔다.

**Router**

URL 정보를 파싱하는 매우 중요한 함수이다.페이지 갱신 요청 마다, gateway를 통해 이 함수를 요청한다.

Router는 다음과 같은 요소를 포함한다.

변수

url [segment]: route함수에 의해 파싱된 url 성분들을 담는 배열이다.

함수

route(): 조건문을 통해 URL 정보에 따라 어떤 컨트롤러가 실행될지 결정하고, 해당 컨트롤러 객체를 생성 후 기본함수를 실행한다.

**Controller**

MVC의 C에 해당하는 요소로, Model과 View를 포함하고 전체적인 로직을 관장하는 객체이다. 모든 Controller는 main 함수를 기본함수로 가진다. 사용자 정의 Controller 객체들은 모두 이 Controller 객체를 상속한다.

Controller는 다음을 포함한다.

변수

url [segment]: segment별로 배열로 url성분을 담고 있는 변수. Router객체의 url을 저장한다.

model [$model]: model 객체들을 배열로 갖는 변수이다. selector는 model객체의 이름 (소문자)이다.

함수

main(): 기본함수.

render ($view): view객체를 호출한다. 화면에 표현하는 기능을 하는 함수다. view객체가 호출되면 view객체가 표현을 담당한다.

request(): url segment 말고GET 파라미터나 POST파라미터로 오는 정보를 처리하는 함수인데, 개발은 추후에 한다.

**Root.php (Controller)**

URL: localhost. (URL규칙에 반하는 예외의 경우이다.)

Root.php 는 다음의 요소를 포함한다.

변수

Model: Term

**Term.php (Controller)**

URL: localhost

변수

Model: Term, Comment

**Model**

후에 언급

**Term.php (Model)**

DB\_Table: term

Term.php 는 다음의 요소를 포함한다.

변수

$data [term, termExact, recentTerm, suggestedTerm, otherDef, addTerm]:

함수

get

**3.3 Performance requirements**

**3.4 Design constraints**

**3.5 Other requirements**