

전자정부표준프레임워크 기반의 Open PaaS 개발

개발 가이드

PHP 애플리케이션

|  |  |  |
| --- | --- | --- |
| 작성자 | 안찬영(인) | 2015.12.14 |
| 검토자 | 송창학(인) |  |
| 승인자 | 현재욱(인) |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

개 정 이 력

| 버전 | 작성일 | 변경내용[[1]](#footnote-2) | 작성자 | 승인자 |
| --- | --- | --- | --- | --- |
| 1.0 | 2015.12.15 | 최초작성 | 안찬영 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

목차

1. 개요
   1. 문서 개요
      1. 목적

본 문서(PHP 애플리케이션 개발 가이드)는 개발형 플랫폼 프로젝트의 서비스를PHP 애플리케이션과 연동하는 방법을 제공하는데 있습니다.

* + 1. 범위

연동을 하는 서비스는 MySQL, MongoDB, Redis, GlusterFS 입니다. 데이터 저장에는 MySQL, MongoDB를 사용하고 사용자의 Session은 Redis에 샘플 어플리케이션에서 사용하는 이미지 파일 관리(Upload)를 위해서는 GlusterFS를 사용합니다.

* + 1. 제약사항

현재 PHP 빌드팩(CloudFoundry의 공식 빌드팩 v4.3.1)의 지원하는 드라이버가 본 사업의 서비스와 정확하게 맞지 않아 일부 서비스(RabbitMQ, CUBRID)는 연동하지 못했습니다. 또한 MongoDB와 연결은 DB인증 절차가 적용이 되지 않아 DB Admin 계정으로 접속하는 방법을 설명하였습니다.

실제 사용시에는 PHP 빌드팩을커스터마이징하여 프로젝트 환경에 맞추어서 개발을 진행하셔야 합니다.

* + 1. 참고 자료
* **PHP 빌드팩 :**<https://github.com/cloudfoundry/php-buildpack>
* **XAMP 사이트 :**<https://www.apachefriends.org/index.html>

1. PHP 애플리케이션개발가이드
   1. 개요

개발환경은 개방형 플랫폼의 네트워크의 구성에 따라 로컬에 구성을 하거나 직접 개방형 플랫폼에 Deploy하여 관리할 수 있습니다. 여기서는 Windows 환경에 간단하게 개발환경을 구성하고 개방형 플랫폼에 배포(Deploy)하는 방법을 설명하겠습니다.

* 1. 개발환경 구성

PHP 개발환경을 구성하려면 Web Server와 PHP 엔진, Extension 설치등을해야하는데 이를 간편하게 구성 놓은 툴이 있습니다. 본 가이드에서는 XAMP를 이용하여 설치를 하고 구성하도록 하겠습니다.

본 문서 작성을 위해 구성한 시스템은 아래와 같습니다.

* OS : Windows 8.1 64bit
* XAMP PHP 5.5.30
* Mongo 라이브러리 :
* Composer :

PHP로 REST/full 서버를 구현하였고 화면(HTML)은 Apache의 Web 서버에서 제공을 합니다. HTML과 PHP는 별도로 돌아가는 구조입니다.

* + 1. PHP 샘플 소스 받기

샘플의 위치는 변경될수있느나 개방형 플랙폼 홈페이지에서 찾아볼 수 있습니다. 해당 GIT 위치를 확인하시고 아래와 같은 명령문으로 소스를 다운로드 받습니다. 해당 명령을 위해서는 GIT Client가 설치되어 있어야 합니다.

|  |
| --- |
| git clone “GIT 저장소 위치” |

* + 1. XAMP설치

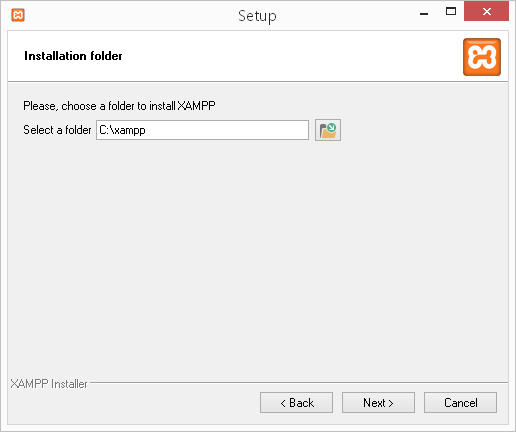
1. URL (<https://www.apachefriends.org/index.html>**)** 에 접속하면 바로 다운로드 화면이 나옵니다. 여기서 “Click here for other version”을 선택합니다.



XAMP 공식 홈페이지 첫화면

다운로드는 Windows버전 PHP 5.5.30 (32bit)을 다운로드 받습니다.

1. 다운로드 받은 파일을 실행하고 모두 Next를 하면됩니다. 하지만 아래와 같이 디렉토리를 물어보는데 이때 이 위치를 변경하거나 정확하게 기억하고 있어야 합니다. 설치 완료후php 실행 디렉토리를 환경변수(Path)에 넣어 줘야 하기 때문입니다.



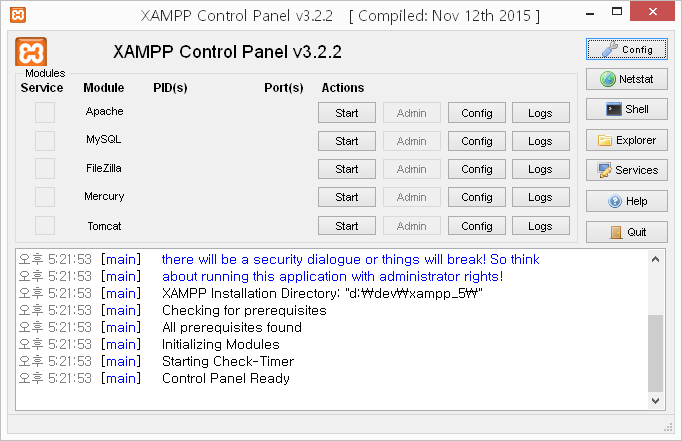
XAMP 설치 디렉토리

1. 설치가 정상적으로 이루어지고 있으면 아래와 같이 진행이 될겁니다. 처음 실행할 때 Antivirus 프로그램으로 느려질수 있다는 문구 등이 나올수도 있는데 “확인”을 누르시면 됩니다.



XAMP 설치 진행중

1. 설치가 완료되면 Control Panel을 띄우겠다는 메시지가 나옵니다. 선택이 Default로 되어 있어 완료를 선택하면 아래와 같은 Control Panel이 실행됩니다.



XAMP 관리 패널 창

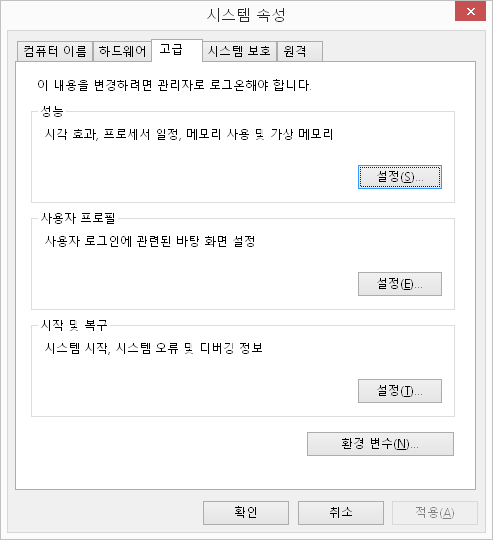
사용방법은 간단합니다. 원하는 서비스(여기서는 Apache 만 사용할 예정)의 Start를 선택하면 해당 서비스가 실행이 됩니다. 단. 해당서비스가 사용하는 포트(Apache의 경우 80)는 사용하고 있지 않아야 합니다.

1. Apache의 Config를 선택하고 Apache (httpd.conf)를 선택하여 DocumentRootdhk Directory의 위치를 개발소스가 설치된 곳으로 바꾸면 브라우져에서<http://localhost> 로 호출시 개발하는 위치로 바로 연결됩니다. 개발소스의 위치는 2.2.1에서 설치한 위치를 지정해 넣습니다.

|  |
| --- |
| DocumentRoot C:\개발소스위치  <DirectoryC:\개발소스위치>  … |

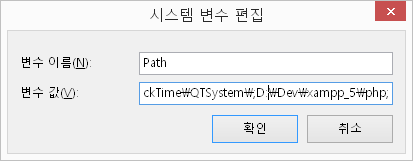
* + 1. PHP 실행 환경설정

1. XAMP로 설치된 PHP를 어디서나 실행가능하게 환경설정(Path)에 넣어줍니다. 제어판 -> 시스템 -> 고급 시스템 설정을 선택하면 아래와 같이 시스템 속성을 변경하는 창이 나타납니다.



시스템 속성 창

1. 여기서 “환경변수”를 선택하고 Path를 편집을 합니다. 변수값의 마지막에 XAMP 설치 디렉토리 아래의 php디렉토리를 추가합니다.



Path 환경변수 설정

1. 정상적으로 구성이 되었는지 확인하려면 “명령 프롬프트”를 실행하고 php–version 을 선택하여 아래와 같이 나오면 정상적으로 환경설정이 완료된 것입니다.



명령 프롬프트에서 PHP 버전 확인

* + 1. Composer 설치

Composer는 개발시 필요한 라이브러리를 관리하는 툴입니다. 홈페이지는 다음과 같습니다. <https://getcomposer.org/>

1. Composer를 다운로드 받아 설치하고 Path에 설정하는 방법이 있지만 여기서는 Manual로 설치를 하겠습니다. Composer.phar 파일을 개방형 플랫폼에서 그대로 사용하고 있어 Manual 설치로 composer.phar파일을 개발 위치에 설치하였습니다. (소스를 Git에서 받았으면 따로 설치할 필요는 없습니다.)
2. 매뉴얼 설치는 간단합니다. 아래와 같이 소스의 루트 디렉토리에서 입력을 하면됩니다. 물론 PHP 명령이 실행이 될 수 있게 2.2.3의 환경변수 설정이 되어 있어야만 합니다.

|  |
| --- |
| php r "readfile('https://getcomposer.org/installer');" | php |

1. 필요한 Package를 composer.json에 구성하고 install하면 PHP에서 사용할 수 있는 Package가 vendor 디렉토리 아래에 설치가 됩니다.

\* 주의사항 \*

개방형 플랫폼에서 사용하기 위해서는 XAMP와는 환경이 틀리기 때문에 PHP 빌드팩의 Release된 Extension을 확인하고 Dependency를 확인해야 합니다. XAMP 환경에서는 동작을 하지만 개방형 플랫폼에서는 동작을 안할 수 있으니 주의해야 합니다.

PHP 빌드팩Extenstion정보 :<https://github.com/cloudfoundry/php-buildpack/releases>

* + 1. Mongo 드라이버 설치

Mongo 드라이브 설치는 개방형 플랫폼에서 지원하는 Mongo 드라이버를 설치하기 위함입니다. 관련문서는 <http://docs.php.net/manual/en/mongo.installation.php#mongo.installation.windows>에 있습니다. 라이브러리 파일을 다운받고 config 파일에 정보를 추가하면 됩니다.

1. 먼저 문서에 PECL 홈페이지(<http://pecl.php.net/package/mongo>)에서 DLL를 다운로드 받아야 합니다. 본 가이드는 1.6.12 버전을 을 선택하였습니다.



위의 링크중 DLL 부분을 눌러서 선택하고 “5.5 Thread Safe (TS) x86”를 다운로드 받습니다. 압축을 해제하면 php\_mongo.dll이 있는데 이 파일만 있으면 됩니다.

1. php\_mongo.dll 파일을 XAMP 설치된 디렉토리에서php 아래에 ext에 복사를 합니다.
2. XAMP 설치 디렉토리의php디렉토리에 php.ini를 선택하고 아래와 같이 추가를 해줍니다.

|  |
| --- |
| extension=php\_mongo.dll |

1. php.ini 파일의 셋팅이 끝났으면 Apache서버를 재기동합니다. 만약에 오류가 있으면 XAMP 패널에 빨간색으로 오류가 표시되니 정상적으로 Apache서버가 올라오는지 확인합니다.
2. 모듈이 정상적으로 설치되었는지 확인하려면 소스코드의 루트에 있는 info.php를 실행합니다. 브라우져에서<http://localhost/info.php> 를 선택하고 내용에 mongo 부분의 설정 정보가 보이면 정상적으로 설치가 된 것입니다.
   1. 개발

개발에 필요한 Package의 구성과 소스디렉토리 구성을 설명합니다. 그리고 각각의 서비스와 연동을 하는 부분의 설명합니다.

* + 1. 사용 Package 설명

Composer를 사용하여 Dependency를 관리합니다. Composer.json 파일의 Package구성을 설명하면 아래의 표와 같습니다. 설정정보는 composer.json에 저장되어 있습니다.

|  |  |  |
| --- | --- | --- |
| 패키지명 | 버전 | 설명 |
| slim/slim | 2.\* | PHP의 REST/full Framewok를 위해 사용합니다. |
| videlalvaro/php-amqplib | 2.5.\* | RabbitMQ 서비스와 연결을 위해 사용합니다. (SSL 접속이 지원되지 않아 현재는 사용하지 못하고 있음) |
| predis/predis | 1.0.\* | Redis 서비스와 연동을 위해 사용합니다. |
| rackspace/php-opencloud | 1.15.\* | Openstack 연결 SDK로 GlusterFS에 파일을 Upload할 때 사용합니다. |
| guzzlehttp/guzzle | 6.\* | Http client로 GlusterFS의 Container 권한 변경시 사용합니다. |
| phpunit/phpunit | 4.3.\* | PHP의 단위테스트를 위한 프로그램입니다. Vendor\bin\phpunit를 실행하면 test 디렉토리의 Test Case를 실행하고 결과를 화면에 표시합니다. |

Composer를 이용하여 라이브러리를 설치하려면 아래와 같이 install을 실행하면 됩니다. 이 명령은 개방형 플랫폼에 PHP 빌드팩과 같이 배포될 때 자동으로 수행이 됩니다.

|  |
| --- |
| phpcomposer.phar install |

만약에 로컬 개발환경에서 composer.json파일이 변경되면 install 대신 update를 이용하여 패키지의 구성을 변경하면 됩니다.

* + 1. 디렉토리설명

개발 편의를 위해 API 서비스를 따로 디렉토리로구성하였으며 resource 디렉토리는 HTML에서 필요한 정적인 파일(js, css, image)이 있습니다.

|  |  |
| --- | --- |
| **파일/폴더** | **목적** |
| .bp-config | PHP 빌드팩에서 사용하는 extension을 정의하는 곳입니다. |
| api | REST API에 대한 소스가 있습니다. 각 서비스별로 구분되어 있습니다. |
| resources | HTML에서 사용하는 정적인 파일(js, css, image 등)이 있습니다. |
| Test | PHP 단위테스트를 위한 테스트 케이스가 있습니다. |
| vendor | Composer로 설치되는 패키지가 있습니다. |
| .cfiignore | 개방형 플랫폼에 배포시 배포 예외를 시키는 파일/디렉토리를 정의합니다. |
| .htaccess | REST/full 구현을 위해 url에 대한 패턴을 정의한 곳입니다. |
| composer.json, composer.phar, composer.lock | Composer파일로 패키지의 Dependency를 관리하는 파일입니다. |
| Info.php | PHP에 설치된 모듈을 확인하기 위한 phpinfo()를 포함한 웹페이지입니다. |
| login.html | 예제 실행을 위한 login 페이지입니다. |
| main.html | 예제의 조직도를 보여주기 위한 main페이지입니다. |
| manage..html | 예제의 데이터를 관리하기 위한 manage 페이지입니다. login이후에 보여지는 화면입니다. |
| manifest.yml | 개방형 플랫폼에 배포(push)할 때 사용하는 설정 파일입니다. |
| phpunit.xml | PHP 단위테스트를 정의한 설정입니다. |

* + 1. 애플리케이션 환경설정

REST/full 서비스를 위한 환경설정과 PHP 빌드팩에서 사용할 Extension을 적용해야 합니다.

1. REST/full 서비스를 위해 .htaccess 구성

REST/full API 형식인 /api/변수 를 처리하기 위한 설정입니다. 루트디렉토리에 .htaccess를 추가하고 아래와 같이 넣어줍니다.

|  |
| --- |
| #<IfModulemod\_rewrite.c>  RewriteEngine On  RewriteCond %{REQUEST\_FILENAME} !-f  RewriteCond %{REQUEST\_FILENAME} !-d  RewriteRule /(.\*)$ /api/api.php?request=$1 [QSA,NC,L]  #</IfModule> |

1. PHP 빌드팩에 Extension 추가

XAMP에서 mongo 드라이브를 추가하였듯이 개방형 플랫폼에도 사용항 Extension 라이브러리를 추가해야합니다. 사용가능한 라이브러리는 여기()에서 확인이 가능합니다.

추가를 위해서는 .bp-config디렉토리에options.json 파일을 만들고 아래와 같이 추가해주면 됩니다.

|  |
| --- |
| {  "PHP\_EXTENSIONS": ["mysqli", "mongo", "amqp"]  } |

* + mysqli :mysql과 연결을 위한 extension
  + mongo : mongo 연결을 위한 extension
  + amqp :rabbitmq와 연결을 위한 extension (현재 SSL연동이 안되서 사용을 못하고 있음)
    1. VCAP\_SERVICES 환경설정 정보

개방형 플랫폼에서는 서비스를 사용하기 위해 서비스 생성/바인딩을 합니다. 그리고 연결된 서비스를 사용하기 위해 VCAP\_SERVICES 환경설정 정보를 가져와야 합니다. 이 정보는 연결할 Host 주소/포트, 사용자명, 비밀번호 등 서비스 접속에 필요한 모든 정보를 포함하고 있습니다.

1. 연결된 서비스 정보 확인하기

CF cli를 통해서 서비스와 연결된 정보를 확인합니다. 이 정보를 직접소스에 넣는 것이 아니라 소스에서 VCAP 정보를 가져와서 초기에 설정정보를 셋팅해야합니다.

|  |
| --- |
| $ cfenvphp-sample  {  "VCAP\_SERVICES": {  "p-mysql": [  {  "credentials": {  "hostname": "10.30.40.63",  "jdbcUrl": "jdbc:mysql://10.30.40.63:3306/cf\_ea68784e\_3de6\_439d\_afc1\_d51b4e95627b?user=ZwCFnQRiT3KANqHZ\u0026password=qs7oqi4nSvWq6UQa",  "name": "cf\_ea68784e\_3de6\_439d\_afc1\_d51b4e95627b",  "password": "qs7oqi4nSvWq6UQa",  "port": 3306,  "uri": "mysql://ZwCFnQRiT3KANqHZ:qs7oqi4nSvWq6UQa@10.30.40.63:3306/cf\_ea68784e\_3de6\_439d\_afc1\_d51b4e95627b?reconnect=true",  "username": "ZwCFnQRiT3KANqHZ"  },  "label": "p-mysql",  "name": "sample-mysql-instance",  "plan": "100mb",  "tags": [  "mysql"  ]  }  ],  …..(이하 생략)….. |

1. PHP에서 환경설정 정보 가져오기

PHP에서 VCAP 환경정보를 가져오는 방법은 간단합니다. System의 env 정보를 가져오는 루틴을 사용하면 됩니다. 아래의 예시는 mysql 서비스의 Conncetion 정보를 가져오는 부분입니다. (위치 :api/mysql\_view.php)

|  |
| --- |
| if (array\_key\_exists("VCAP\_SERVICES", $\_ENV)) {  // $\_ENV (시스템 환경설정) 정보에서 VCAP Services가 있는지를 체크합니다.  $env = json\_decode($\_ENV["VCAP\_SERVICES"], $assoc=true);  //VCAP\_SERVICES의 내용을 JSON 객체로 컨버젼합니다.  $this->host = $env["p-mysql"][0]["credentials"]["hostname"].':'$env["p-mysql"][0]["credenti  alls"]["port"];  // host위치정보와 port정보를 가져옵니다.  $this->username = $env["p-mysql"][0]["credentials"]["username"];  // 사용자정보를 가져옵니다.  $this->password = $env["p-mysql"][0]["credentials"]["password"];  // 사용자 비밀번호 정보를 가져옵니다.  $this->dbname = $env["p-mysql"][0]["credentials"]["name"];  // 사용자의 DB 명을 가져옵니다. |

환경설정정보는 서비스마다 위치/명칭이 다릅니다. cfenv 명령으로 정확한 위치를 파악하거나 서비스 제공자의 매뉴얼을 참조하여야 합니다.

* + 1. Mysql 연동

Extenstion에 추가한 mysqli를 사용합니다. XAMP에서는 기본으로 설정이 되어 있기 때문에 따로 설치작업이 필요없습니다.

(위치 :api/mysql\_view.php)

1. Mysql 에 접속하기

|  |
| --- |
| $conn = new mysqli($this->host, $this->username, $this->password, $this->dbname);  if($conn->connect\_error) {  die("conncetion failed:".$conn->connect\_error);  } |

mysqli를 이용하여 mysql 서비스에 접속합니다. 환경설정에서 가져온 host, username, password와 db명을 이용하여 접속을 합니다.

1. Query보내고 결과값 받기

Query를 작성하고 Prepared Statement로 실행을 합니다. 실행된 결과값을 받아서 원하는 형태의 Array로 만들어 줍니다. 모든 처리가 완료되면 close로 connection과 statement를 종료합니다.

|  |
| --- |
| $sql = "SELECT \* FROM ORG\_TBL";  $stmt = $conn->prepare($sql);  $stmt->execute();  $org\_result = $stmt->get\_result();  if ($org\_result->num\_rows> 0) {  while($row = $org\_result->fetch\_assoc()) {  $org = array(  "id" =>strval($row["id"]),  "label" => $row["label"],  "desc" => $row["desc"],  "url" => $row["url"],  "created" => $row["created"],  "modified" => $row["modified"]  );  array\_push($result["orgs"], $org);  }  $result["org"] = $org;  }  $stmt->close();  $conn->close(); |

1. 결과값을 Json으로컨버전하여 HTML에서 처리할 수 있도록 합니다.

|  |
| --- |
| echo json\_encode($result); |

* + 1. CUBRID 연동

현재 CF의 기본 빌드팩에서는 CUBRID를 지원하지 않아 본 샘플에서는 구현하지 않았습니다. 만약 프로젝트에서 CUBRID를 사용해야하면 별도로 문의 바랍니다.

* + 1. MongoDB 연동

Extenstion에 추가한 mongo 라이브러리를 이용합니다. 단 현재 mongo 라이브러리로는 사용자 인증에 문제가 있습니다. 라이브러리가 버그 fix가 되어야 합니다. 본 가이드에서는 부득이하게 MongoDB의 Root계정으로 접속하여 예제를 구현하였습니다.

(위치 :api/mongodb\_view.php)

1. Mongodbl 에 접속합니다. 환경설정에서 받아온 uri 정보를 이용합니다.

|  |
| --- |
| $mongo = new MongoClient($this->uri); |

1. Collection을 설정하고 해당 Collection에서 정보를 요청합니다. Find 명령을 이용하여 필요한 정보를 요청합니다. 받아온 결과는 $cursor에 넣고 원하는 데이터 형태로 변경합니다.

|  |
| --- |
| $collection = $mongo->selectCollection($this->dbname, 'ORG\_TBL');  $cursor = $collection->find(array('\_id'=>new MongoId($org\_id)));  foreach ($cursor as $row) {  $org = array(  "id" =>strval($row["\_id"]),  "label" =>isset($row["label"]) ? $row["label"] : "",  "desc" =>isset($row["desc"]) ? $row["desc"] : "",  "url" =>isset($row["url"]) ? $row["url"] : "",  "created" =>isset($row["created"]) ? $row["created"] : "",  "modified" =>isset($row["modified"]) ? $row["modified"] : ""  );  $result["org"] = $org;  } |

1. 결과값을 Json으로컨버전하여 HTML에서 처리할 수 있도록 합니다.

|  |
| --- |
| echo json\_encode($result); |

* + 1. Redis 연동

Redis 연동은 추가로 Composer를 통해 설치가된 패키지를 사용합니다.

1. Redis 사용을 위해서 Predis의 Class에서 register를 선언합니다.

|  |
| --- |
| Predis\Autoloader::register(); |

1. Redis에 접속을 합니다. 환경설정에서 받은 Host, Port, Passworf를 이용하여 Redis에 접속을 합니다.

|  |
| --- |
| $redis = new Predis\Client(  array(  "scheme" => "tcp",  "host" => $this->host,  "port" => $this->port,  "password" => $this->password  )); |

1. Session key와 사용자 ID(username)을 Redis에 저장합니다.

|  |
| --- |
| $redis->set($key, $username); |

* + 1. RabbitMQ연동

CF의 PHP 빌드팩에서amqp접속시 SSL 접속에 문제가 있습니다. 그래서 해당 서비스 연동은 구현이 안되어 있습니다. 접속방법만 명시한 php 파일만 있습니다. (위치 :api/rebbitmq\_view.php)

* + 1. GlusterFS 연동

php-opencloud라는 패키지를 사용하며 composer를 통해서 설치가 되게 되어 있습니다. 단 Container를 Public하게 생성하는 SDK가 없어서 API를 직접 호출(REST형식)하여 권한을 Public으로 설정하고 있습니다.

1. GlusterFS와 연동하기 (파일 Upload)

개방형 플랫폼에서는 Object Storage를 GlusterFS를 사용하는데 Object Storage를 API를 통해 사용하기 위해 Openstack의 Swift를 이용하여 서비스를 할 수 있게 구성되어 있습니다.

php-opencloud는 swif를 만든 rackspace 회사에서 제공하는 SDK입니다.

Opencloud를 사용하기 위해 선언을 합니다.

|  |
| --- |
| use OpenCloud\Rackspace; |

1. Openstack(Object Storage)에 접속을 하고 잘 접속이 되었는지 체크합니다.

|  |
| --- |
| $client = new OpenCloud\OpenStack($this->host, array(  "username" => $this->username,  "password" => $this->password,  "tenantName" => $this->tenantName,  ));  $client->authenticate(); |

1. 파일을 올리기 위한 Container를 설정합니다. 만약에 해당 Container(directory)가 없으면 Container를 새로 생성을 합니다. 그리고 생성된 Container는 Public(읽기권한)으로 설정하여 인증없이 모든 사람이 읽을 수 있게 합니다.

|  |
| --- |
| $service = $client->objectStoreService($this->catalogName, 'RegionOne', 'publicURL');  $container;  // Container를 가져오기  try {  $container = $service->getContainer($this->containerName  } catch (Exception $e) {  // 생성하기  $container = $service->createContainer($this->containerName);  // 만들어진 Container를 Public 모드로 변경하기  // PHP-OpenCloud SDK에서 해당 부분을 지원하지 않아서 직접 API를 호출하여 설정함  $baseUrl = $container->getService()->getEndpoint()->getPublicUrl().'/'.$this->containerName;  $httpClient = new GuzzleHttp\Client();  $res = $httpClient->request('POST', $baseUrl,  ['headers' => ['X-Auth-Token' => $container->getService()->getClient()->getToken(), 'X-Container-Read' => '.r:\*']]  );  // Response Code가 204로 넘어옴. 성공  } |

1. 파일을 해당 컨테이너에 Upload합니다. Container를 Public으로 설정하였기 때문에 이미지의 URL만 있으면 어디서든 읽어올 수 있습니다.

|  |
| --- |
| $fileName = time().'\_'.$fileName;  // 파일 저장  $result = $container->uploadObject($fileName, fopen($file, 'r+'), array('name'=> $fileName, 'Content-Type' => 'image/jpeg')); |

1. 저장된 결과를 URL 형태로 만듭니다. 이 URL로 직접 접속하여 해당 이미지에 접근이 가능합니다.

|  |
| --- |
| $result = array('thumb\_img\_path' => $container->getService()->getEndpoint()->getPublicUrl().'/'.$this->containerName.'/'.$fileName); |

1. 결과값을 Json으로컨버전하여 HTML에서 처리할 수 있도록 합니다.

|  |
| --- |
| echo json\_encode($result); |

* 1. 배포

개발형 플랫폼에 샘플 애플리케이션을 설치하기 위한 부분입니다. CF PUSH 명령문을 사용하기 위한 사전작업과 서비스를 생성하고 연결하는 작업을 설명하고 있습니다.

1. ./manifest.yml 생성

* cf push 명령시 현재 디렉토리의manifest.yml을 참조하여 배포가 진행된다.

|  |
| --- |
| ---  applications:  - name:php-sample-app**# 애플리케이션 이름**  memory: 128M**#** 애플리케이션 **메모리 사이즈**  instances: 1**# 애플리케이션 인스턴스 개수**  path: .  buildpack: https://github.com/cloudfoundry/php-buildpack.git**# 사용할 빌드팩을 선언** |

※애플리케이션 스테이징시할달 받은 포트가 환경변수로 등록되어있다. 이 포트는 애플리케이션의 상태 체크에도 사용되므로 위와 같이 포트를 지정할 것을 권장한다.

1. 개방형 플랫폼 로그인

|  |
| --- |
| **$ cfapi https://api.cf.open-paas.com# 개방형 플랫폼 TARGET 지정**  **#cfapi [target url]**  **$ cf login -u testUser -o sample\_test -s sample\_space# 로그인 요청**  **#cf login –u [user name] –o [org name] –s [space name]**  API endpoint: https://api.cf.open-paas.com  Password>  Authenticating...  OK  Targeted org sample\_test  Targeted space sample\_space  API endpoint: https://api.cf.open-paas.com (API version: 2.29.0)  User: testUser  Org: sample\_test  Space: sample\_space  $ |

1. 개방형 플랫폼 서비스 생성

|  |
| --- |
| **$ cf marketplace# 마켓플레이스 목록 요청**  **service plans description**  p-mysql 100mb, 1gb MySQL databases on demand  p-rabbitmqstandard RabbitMQ is a robust …..  redis-sb shared-vm, dedicated-vm Redis service to provide a ……  **$ cf create-service p-mysql 100mb sample-mysql-instance# 서비스 생성**  **#cf create-service [service] [plan] [service name]**  **$ cf services# 서비스 목록 조회**  **nameservice plan bound apps last…**  sample-mysql-instance p-mysql100mb node-sample, p.... …  sample-rabbitmq-instance p-rabbitmq standard python-sample-.... …  sample-redis-instanceredis-sbshared-vmpython-sample-.... …  $ cf bind-servicephp-sample-app sample-mysql-instance**# 애플리케이션 서비스 바인딩**  **#cf bind-service [app name] [service name]**  **$ cf start php-sample-app# 애플리케이션 시작**  **#cf start [app name]** |

1. 개방형 플랫폼 애플리케이션에 서비스 바인딩 및 애플리케이션 시작

|  |
| --- |
| **$ cf push --no-start**  **# 애플리케이션 업로드만 실행하고 시작하지는 않는다.**  **$ cf services# 서비스 목록 조회**  **nameservice plan bound apps last…**  sample-mysql-instance p-mysql100mb node-sample, p.... …  sample-cubrid-instanceCubridDButf8 node-sample, p.... …  sample-mongo-instanceMongo-DB default-plan node-sample, p.... …  sample-rabbitmq-instance p-rabbitmq standard python-sample-.... …  sample-redis-instanceredis-sbshared-vmpython-sample-.... …  sample-glusterfs-instanceglusterfsglusterfs-1000Mb glusterfs-samp.... …  **$ cf bind-service php-sample-app sample-mysql-instance# 애플리케이션 서비스 바인딩**  **$ cf start php-sample-app# 애플리케이션 시작** |

1. Mysql, Cubrid 테이블 생성

* Sample App의 조직관리 기능을 위해 DB에 테이블을 생성해 주어야 한다.
* Mysql과 Cubrid에 테이블을 추가하는 방법은 OpenPaaS Mysql, Cubrid 서비스팩 설치 가이드의 'Client 툴 접속'을 참고한다.
* Client 툴을 이용하여 아래의 테이블 생성 sql를 각각 실행한다. (Mysql과 Cubrid 양쪽다 동일한 sql로 생성가능하다.)

|  |
| --- |
| DROP TABLE IF EXISTS ORG\_TBL;  DROP TABLE IF EXISTS GROUP\_TBL;  CREATE TABLE ORG\_TBL (  id INT AUTO\_INCREMENT PRIMARY KEY  , label VARCHAR(40) NOT NULL  , `desc` VARCHAR(150)  , url VARCHAR(500) DEFAULT '#'  , created TIMESTAMP DEFAULT CURRENT\_TIMESTAMP NOT NULL  , modified TIMESTAMP  );  CREATE TABLE GROUP\_TBL (  id INT AUTO\_INCREMENT PRIMARY KEY  , org\_id INTEGER NOT NULL  , parent\_id INTEGER  , label VARCHAR(40) NOT NULL  , `desc` VARCHAR(150)  , thumb\_img\_name VARCHAR(256)  , thumb\_img\_path VARCHAR(512)  , url VARCHAR(500) DEFAULT '#'  , created TIMESTAMP DEFAULT CURRENT\_TIMESTAMP NOT NULL  , modified TIMESTAMP  );  ALTER TABLE GROUP\_TBL  ADD FOREIGN KEY(org\_id)  REFERENCES ORG\_TBL(id)  ON DELETE CASCADE;  ALTER TABLE GROUP\_TBL  ADD FOREIGN KEY(parent\_id)  REFERENCES GROUP\_TBL(id)  ON DELETE CASCADE; |

* 1. 테스트

PHP 단위테스트는 phpunit를 이용해서 합니다. 테스트 케이스는 test 디렉토리에 있으며 단위 테스트 실행은 아래와 같이 실행합니다.

|  |
| --- |
| Vendor\bin\phpunit |

1. 변경 내용: 변경이 발생되는 위치와 변경 내용을 자세히 기록(장/절과 변경 내용을 기술한다.) [↑](#footnote-ref-2)