
Symfony2 の勉強 ToDoList アプリ

2013.05.25 渋谷の VOYAGE GROUP AJITO で行われた symfony2 勉強会のワークショップを通して習ったことを復習。xampp のインストールからアプリ完成までの過程をメモ。

1. 環境

OS : Windows XP Professional SP3

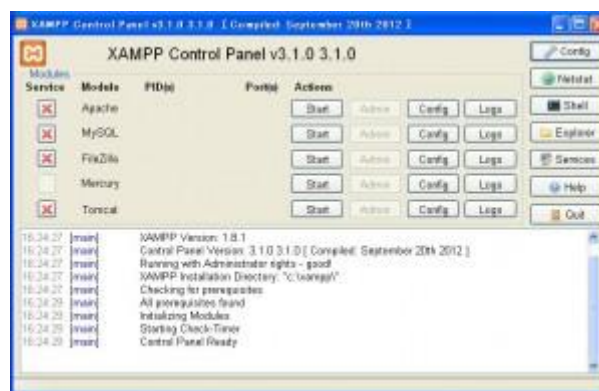
xampp v3.1.0 : c:\xampp にインストール

Symfony 2.2.1 : c:\xampp\htdocs\Symfony にインストール

Google Chrome 27.0.1453.94 m

2. xampp をダウンロードしてインストール

<http://www.apachefriends.org/en/xampp-windows.htm>



apache がスタートでない場合、80 番ポートが占有されている可能性がある、参考資料：
「80 番ポートが何かに占有されている場合の対策」を参照

3. composer をインストール

3.1 composer-setup.exe をダウンロード

<http://getcomposer.org/download/>

ダウンロードしたインストーラを実行したところ、下記 Warning が発生した。

Some settings on your machine may cause stability issues with Composer.
If you encounter issues, try to change the following:

The openssl extension is missing, which will reduce the security and stability of Composer.
If possible you should enable it or recompile php with --with-openssl

3.2 composer インストール時の Warning の解消

下記 URL を参照

<http://stackoverflow.com/questions/2643462/setting-up-ssl-on-a-local-xampp-apache-server>

1.C:\xampp\apache\conf\httpd.conf の以下の部分を変更

```
LoadModule ssl_module modules/mod_ssl.so
```

(変更後)頭に#があれば削除

2.C:\xampp\php の以下の部分を変更

```
extension=php_openssl.dll
```

(変更後) 頭に;があれば削除

再度 composer-server.exe を実行したが composer.phar がインストールされない・・・
その場合次へ・・・

3.3 composer.phar をインストール

ターミナルから下記を実行。php フォルダ配下に composer.phar がインストールされる。

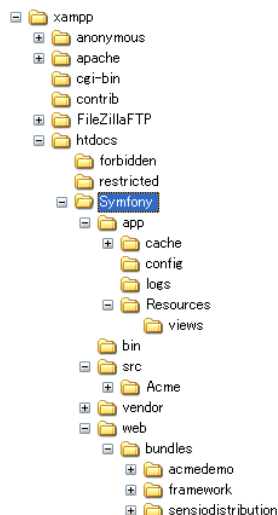
```
C:\xampp\php>php -r  
"eval('>'.file_get_contents('https://getcomposer.org/installer'))";"
```

このコードが何をしているかは参考資料 : 「`php -r "eval('>'.file_get_contents('https://getcomposer.org/installer'))';"`とは」を参照

4. symfony をインストール

4.1 composer を使って symfony をインストールする。

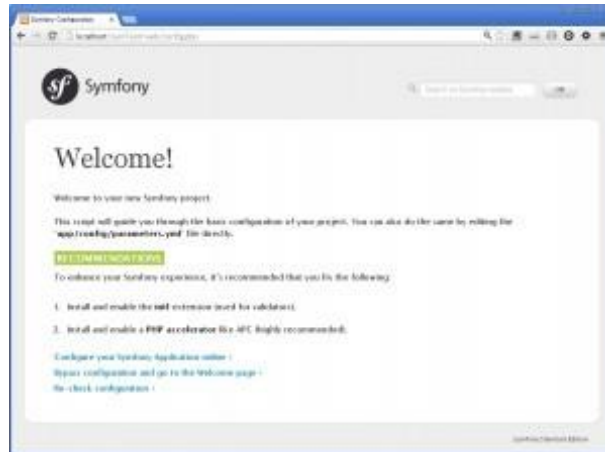
```
C:\xampp\php>php composer.phar create-project  
symfony/framework-standard-edition ../htdocs/Symfony 2.2.1
```



5. 事前準備 : apache、php 環境設定

5.1 apache を起動し、symfony の welcome ページにアクセス

<http://localhost/symfony/web/config.php>



画面中ほどの RECOMMENDATIONS(お勧め設定)を確認し対応する。次へ・・・

To enhance your Symfony experience, it's recommended that you fix the following: 1. Install and enable the intl extension (used for validators). 2. Install and enable a PHP accelerator like APC (highly recommended).

5.2 intl extension を有効にする

intl とは *Unicode* 文字列を扱ったり、国際化対応するための拡張機能

xampp のコンソール上で、Appache の【Config】ボタンをクリックすると、下図のようになり、PHP.ini を開くことが出来る。



php_intl.dll の記述を変更する

```
;extension=php_intl.dll(変更前)  
extension=php_intl.dll(変更後)
```

5.3 PHP accelerator をインストールする

PHP のコードをコンパイルすることで *PHP* の動作速度を向上させる機能

1. PHP のバージョンと、Zend の確認

<http://localhost/xampp/phpinfo.php>



PHP Version =5.4.7

Zend Extension Build = API220100525,TS,VC9

2. 対応する PHP accelerator をダウンロード

<http://windows.php.net/downloads/pecl/releases/apc/>

一番新しい項目を選択する。

windows.php.net - /downloads/pecl/releases/apc/

To Parent Directory			
Monday, August 26, 2013 2:25 PM	<dir>	3.1.14	
Monday, August 26, 2013 2:25 PM	<dir>	3.1.13	
Thursday, November 14, 2012 10:08 PM	<dir>	3.1.12	
Thursday, January 03, 2013 12:16 PM	<dir>	3.1.11	

「5.4」「ts」を頼りに、「php_apc-3.1.14-5.4-ts-vc9-x86-xp.zip」をダウンロード。vista の場合は php_apc-3.1.14-5.4-ts-vc9-x86.zip こっちかな。。

windows.php.net - /downloads/pecl/releases/apc/3.1.14/

To Parent Directory			
Thursday, January 03, 2013 11:49 AM	210871	php_apc-3.1.14-5.4-ts-vc9-x86-xp.zip	
Thursday, January 03, 2013 11:49 AM	208488	php_apc-3.1.14-5.4-ts-vc9-x86-xp.zip	
Thursday, January 03, 2013 11:49 AM	212356	php_apc-3.1.14-5.4-ts-vc9-x86-xp.zip	
Thursday, January 03, 2013 11:49 AM	211948	php_apc-3.1.14-5.4-ts-vc9-x86-xp.zip	
Thursday, January 03, 2013 11:50 AM	210182	php_apc-3.1.14-5.4-ts-vc9-x86-xp.zip	
Thursday, January 03, 2013 11:50 AM	210871	php_apc-3.1.14-5.4-ts-vc9-x86-xp.zip	
Thursday, January 03, 2013 11:50 AM	210871	php_apc-3.1.14-5.4-ts-vc9-x86-xp.zip	
Thursday, January 03, 2013 11:50 AM	210871	php_apc-3.1.14-5.4-ts-vc9-x86-xp.zip	
Thursday, January 03, 2013 11:50 AM	210871	php_apc-3.1.14-5.4-ts-vc9-x86-xp.zip	
Thursday, January 03, 2013 12:16 PM	210871	php_apc-3.1.14-5.4-ts-vc9-x86-xp.zip	

3. ダウンロードしたファイルに含まれる「php_apc.dll」を下記フォルダに移動

C:\xampp\php\ext

4. PHP.ini に下記 1 行を追加する。

extension=php_apc.dll

5.4 symfony2 のプロジェクト作成終了



xamppを起動しapacheをスタートすると、“icuuc46.dll”が無いというエラーが出る場合がある。xampp のバグらしいので、下記を実行。

- Go to [xampp_path]/php/
- Select the following 3 files:
 - icudt46.dll
 - icuin46.dll
 - icuuc46.dll
- Copy
- Paste the files in [xampp_path]/apache/bin folder.

(参考) <http://cynwong.com/blog/?p=446>

6. ToDoList 用のプロジェクトを作成する。

6.1 バンドル生成

新規に Symfony のプロジェクト（バンドル）を生成。

```
C:\xampp\htdocs\Symfony>php app/console generate:bundle  
--namespace=My/ToDoListBundle --format=annotation
```

この作業は Symfony のルートディレクトリ直下で行う必要がある。

src/My/ToDoListBundle というフォルダが出来できる。

--format=annotation なぜこれかは不明。マニュアルには--format=yml もある。

上記を実行すると、下記のメッセージが表示されるので Enter、Yes、No を入力。

```
Welcome to the Symfony2 bundle generator
```

```
In your code, a bundle is often referenced by its name. It can  
be the concatenation of all namespace parts but it's really up  
to you to come up with a unique name (a good practice is to start  
with the vendor name).  
Based on the namespace, we suggest MyToDoListBundle.
```

```
Bundle name [MyToDoListBundle]:(enter)
```

```
The bundle can be generated anywhere. The suggested default  
directory uses the standard conventions.
```

```

Target directory [C:/xampp/htdocs/Symfony/src]:(enter)
To help you get started faster, the command can generate some
code snippets for you.
Do you want to generate the whole directory structure [no]? (yes)

Summary before generation

You are going to generate a "MyToDoListBundle" bundle
in "C:/xampp/htdocs/Symfony/src/" using the "annotation"
format.
Do you confirm generation [yes]? (yes)

Bundle generation

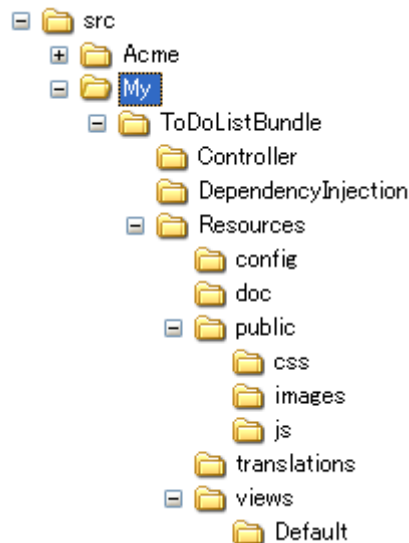
Generating the bundle code: OK
Checking that the bundle is autoloaded: OK
Confirm automatic update of your Kernel [yes]? (yes)
Enabling the bundle inside the Kernel: OK
Confirm automatic update of the Routing [yes]? (yes)
Importing the bundle routing resource: OK

You can now start using the generated code!

```

Symfony/srcの下に My/ToDoListBundle が生成される。

whole directory structure?=yes にしたのは後で Resources/public の下に Twitter bootstrap を導入する為。



7. POST エンティティを生成

generate:doctrine:entity コマンドを使って、post エンティティを生成する。

```

C:\xampp\htdocs\Symfony>php app/console
generate:doctrine:entity --entity=MyToDoListBundle:Post
--format=annotation --fields="title:string(255) note:text
deadline:text createdAt:datetime updatedAt:datetime"

```

generate:doctrine:entity : doctrine はデータベースを扱うためのライブラリ

--entity=MyToDoListBundle:Post : ショートカット名が MyToDoListBundle:Post

```
--format=annotation : yml, xml, php、annotation が使える
--fields=" : フィールドを定義
title:string(255) : title フィールドは文字列 255 文字。項目のタイトルを書く。
note:text : note フィールド。ここに内容を書く。
deadline:text : deadline フィールドには締切日を書く。
createdAt:datetime : 作成日を日付型で保存するフィールド
updatedAt:datetime : 更新日を日付型で保存するフィールド
"
```

上記を実行すると、下記のメッセージが表示されるので Enter、Yes、No を入力

```
Welcome to the Doctrine2 entity generator

This command helps you generate Doctrine2 entities.
First, you need to give the entity name you want to generate.
You must use the shortcut notation like AcmeBlogBundle:Post.

The Entity shortcut name [MyToDoListBundle:Post]: (enter)

Determine the format to use for the mapping information.
Configuration format (yaml, xml, php, or annotation) [annotation]:
(enter)

Instead of starting with a blank entity, you can add some fields
now.
Note that the primary key will be added automatically (named id).
Available types: array, simple_array, json_array, object,
boolean, integer, smallint, bigint, string, text, datetime,
datetimetz, date, time, decimal, float, blob, guid.

New field name (press <return> to stop adding fields): (enter)

Do you want to generate an empty repository class [no]? (yes)

Summary before generation

You are going to generate a "MyToDoListBundle:Post" Doctrine2
entity using the "annotation" format.

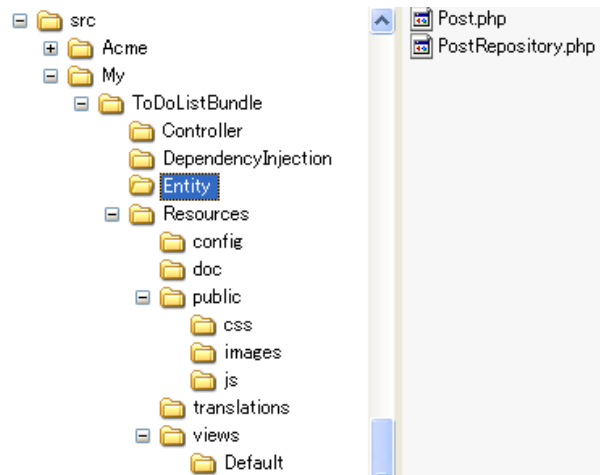
Do you confirm generation [yes]? (yes)

Entity generation

Generating the entity code: OK

You can now start using the generated code!
```

src/My/ToDoListBundle/Entity フォルダと、その中に 2 ファイルが生成された。



8. ToDoList を保存する為のデータベース&テーブル作成

8.1 parameters.yml の設定を確認

mysql の中に、symfony という名前の DB が出る。user は root。
確認するだけにする。

```
parameters:
  database_driver:  pdo_mysql
  database_host:    127.0.0.1
  database_port:    ~
  database_name:    symfony
  database_user:    root
  database_password: ~
```

C:\xampp\htdocs\Symfony\app\config\parameters.yml

8.2 Entity フォルダ内のファイルを確認

Post エンティティの生成によって、Entity フォルダとその中にファイルが 2 つ作成される。

```
～省略～
class Post
{
  ～省略～
  /**
   * @var string
   * @ORM\Column(name="title", type="string", length=255)
   */
  private $title;

  /**
   * @var string
   * @ORM\Column(name="note", type="text")
   */
  private $note;
```

```
/**
 * @var string
 *
 * @ORM\Column(name="deadline", type="text")
 */
private $deadline;
```

～省略～

C:\xampp\htdocs\Symfony\src\My\ToDoListBundle\Entity\Post.php

～省略～

```
class PostRepository extends EntityRepository
{
}
```

～省略～

src\My\ToDoListBundle\Entity\PostRepository.php

8.3 データベースを生成

xampp の control panel を起動し MySQL をスタートさせる。

その後、ターミナルで下記を実行する。symfony という DB が生成される。

```
C:\xampp\htdocs\Symfony>php app/console
doctrine:database:create
Created database for connection named `symfony`
```

その後、ターミナルで下記を実行する。symfony の中にテーブルが生成される。

```
C:\xampp\htdocs\Symfony>php app/console doctrine:schema:create
ATTENTION: This operation should not be executed in a production
environment.

Creating database schema...
Database schema created successfully!
```

Post.php と PostRepository.php を認識して生成されているのだろう・・・

symfony が存在するかを確認。

```
C:\xampp\htdocs\Symfony>mysql -u root -p
Enter password:
```

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| cdcol |
| mysql |
| performance_schema |
| phpmyadmin |
| symfony |
| test |
| webauth |
+-----+
8 rows in set (0.06 sec)
```

テーブルを確認

```
mysql> use symfony;
Database changed
mysql> show tables;
```

```
+-----+
| Tables_in_symfony |
+-----+
| post              |
+-----+
1 row in set (0.00 sec)
```

8.4 データベースにレコードを 1 つ入れておく

```
mysql> INSERT INTO Post (title, note, deadline, createdAt,
updatedAt) values ('Symfony2 の勉強', 'Symfony2 の勉強を通して
ToDoList アプリを作ってみる.', 'ASAP', NOW(), NOW());
Query OK, 1 row affected (0.03 sec)
```

9. ToDoList 画面作成

9.1 コントローラ作成

routing を行うため controller に annotation を設定。

```
～省略～

/**
 * @Route("/todo")
 * @Template()
 */
class ToDoListController extends Controller
{
    /**
     * @Route("/")
     */
    public function indexAction()
    {
        $em = $this->get('doctrine')->getManager();
        $posts =
        $em->getRepository('MyToDoListBundle:Post')->findAll();

        return array('posts' => $posts);
    }
}
```

src/My/ToDoListBundle/Controller/ToDoListController.php

* @Route("/todo")

localhost:8000/app_dev.php/todo/にアクセスがあった場合、この annotation を見つけて実行するというルーティングを行っている。

indexAction()の index 部分と、index.html.twig の index 部分がマッチングされる。

ToDoListController は変えられない。

9.2 テンプレート作成

下記のファイルを作成し保存。

```
<h1>My To Do Items</h1>
<table class="table">
```

```

<thead>
  <tr>
    <td>ID</td>
    <td>登録日</td>
    <td>タイトル</td>
    <td>締切日</td>
  </tr>
</thead>
<tbody>
  {% posts 配列をループして、投稿記事の情報を表示 %}
  {% for post in posts %}
    <tr>
      <td>{{ post.id }}</td>
      <td>{{ post.createdAt|date('Y/m/d H:i') }}</td>
      <td>{{ post.title }}</td>
      <td>{{ post.deadline }}</td>
    </tr>
  {% else %}
    <tr>
      <td colspan="3">nothing to do</td>
    </tr>
  {% endfor %}
</tbody>
</table>

```

src/My/ToDoListBundle/Resources/views/ToDoList/index.html.twig

views/ToDoList/index.html.twig **ここも変えられない。**

9.3 確認

ビルトインサーバを起動する。

```
C:\xampp\htdocs\Symfony>php -S localhost:8000 -t web
```

http://localhost:8000/app_dev.php/todo/

ブラウザで上記 URL にアクセスすると下記の様な表示がされる。

My To Do Items

ID	登録日	タイトル	締切日
1	2013/05/29 11:01	Symfony2の勉強	ASAP

10. 詳細内容画面作成

10.1 コントローラ追加

内容表示画面用のコントローラを追加

～省略～

```

/**
 * @Route("/{id}/detail")

```

```

    */
    public function detailAction($id)
    {
        $em = $this->get('doctrine')->getManager();
        $post =
        $em->getRepository('MyToDoListBundle:Post')->find($id);

        if (!$post) {
            throw $this->createNotFoundException('The note does not
            exist');
        }
        return array('post' => $post);
    }

```

～省略～

src/My/ToDoListBundle/Controller/ToDoListController.php

detailAction の detail 部分と、detail.html.twig の detail 部分がマッチングされる。

10.2 テンプレート作成

```

<h1>{{ post.title }}</h1>

<p><small>Created: {{ post.createdAt|date('Y/m/d
H:i') }}</small></p>

<p><small>Updated: {{ post.updatedAt|date('Y/m/d
H:i') }}</small></p>

<p>{{ post.note|nl2br }}</p>

<p><small>Deadline is {{ post.deadline|nl2br }}</small></p>

```

src/My/ToDoListBundle/Resources/views/ToDoList/detail.html.twig

10.3 確認

http://localhost:8000/app_dev.php/todo/1/detail

ブラウザで上記 URL にアクセスすると下記の様な表示がされる。

Symfony2の勉強

Created: 2013/05/29 11:01

Updated: 2013/05/29 11:01

Symfony2の勉強を通してToDoListアプリを作ってみる。

Deadline is ASAP

11. ToDoList 画面の各要素と詳細内容画面をリンクさせる

11.1 @Route に名前を設定する

@Route に名前をつけると、テンプレート内でリンクする事ができる。

～省略～

```

class ToDoListController extends Controller

```

```
{
    /**
     * @Route("/", name="todo_index")
     */

    ~省略~

    /**
     * @Route("/{id}/detail", name="todo_detail")
     */
}
```

src/My/ToDoListBundle/Controller/ToDoListController.php

11.2 テンプレートで@Root の name を使って、記事一覧画面の各要素と、記事詳細画面とのリンクを張る。

タイトル部分にリンクを張る

```
~省略~

<td>{{ post.id }}</td>
<td>{{ post.createdAt|date('Y/m/d H:i') }}</td>
<td><a href="{{ path('todo_detail', {'id':
post.id}) }}">{{ post.title }}</a></td>
<td>{{ post.deadline }}</td>

~省略~
```

src/My/ToDoListBundle/Resources/views/ToDoList/index.html.twig

My To Do Items

ID	登録日	タイトル	締切日
1	2013/05/29 11:01	Symfony2の勉強	ASAP

記事詳細画面の一番下に「一覧に戻る」リンクを作成

```
~省略~

<p><small>Deadline is {{ post.deadline|n12br }}</p>

<p><a href="{{ path('todo_index') }}">一覧に戻る</a></p>
```

src/My/ToDoListBundle/Resources/views/ToDoList/detail.html.twig

Symfony2の勉強

Created: 2013/05/29 11:01

Updated: 2013/05/29 11:01

Symfony2の勉強を通してToDoListアプリを作ってみる。

Deadline is ASAP

[一覧に戻る](#)

12. テンプレートの継承

12.1 継承元（親テンプレート）の作成

view 直下に置く。

```
<!doctype html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html;
  charset=utf-8" />

  <script type="text/javascript"
  src="//ajax.googleapis.com/ajax/libs/jquery/1.8/jquery.min.js"></script>

  <title>{% block title %}{{ block('page_title') }} - My to do
  list{% endblock title %}</title>
</head>
<body>
  <div>
    <div>
      <div>
        <h1>My to do list.</h1>
      </div>

      <div>
        <h2>
          {% block page_title %}{% endblock page_title %}
        </h2>
        {% block content %}{% endblock content %}
      </div>
    </div>

    <footer>
      <p>*** テンプレート(base.html)を継承 ***</p>
    </footer>
  </div>

  {% block javascripts %}{% endblock javascripts %}
</body>
</html>
```

src¥My¥ToDoListBundle¥Resources¥views¥base.html.twig

12.2 ToDoList 画面の変更

- 1.頭に{% extends 'MyToDoListBundle::base.html.twig' %}を挿入。
2. <h1> </h1>を削除し、代わりに{% block page_title 'Items' %}を挿入。
- 3.元の内容を{% block content %}{% endblock content %}で囲む。

```
{% extends 'MyToDoListBundle::base.html.twig' %}

{% block page_title 'Items' %}

{% block content %}

<table class="table">

～省略～

</table>

{% endblock content %}
```

src¥My¥ToDoListBundle¥Resources¥views¥ToDoList¥index.html.twig

12.3 詳細表示画面の変更

ToDoList 画面と同様の処理をする

```
{% extends 'MyToDoListBundle::base.html.twig' %}

{% block page_title %}{% post.title %}{% endblock %}

{% block content %}

<p><small>Created: {% post.createdAt|date('Y/m/d
H:i') %}</small></p>

<p><small>Updated: {% post.updatedAt|date('Y/m/d
H:i') %}</small></p>

<p>{% post.note|nl2br %}</p>

<p><small>Deadline is {% post.deadline|nl2br %}</small></p>

<p><a href="{% path('todo_index') %}">一覧に戻る</a></p>

{% endblock content %}
```

src¥My¥ToDoListBundle¥Resources¥views¥ToDoList¥detail.html.twig

12.4 確認

ブラウザで上記 URL にアクセスすると下記の様な表示がされる。base.html.twig に定義したヘッダーとフッターが継承されている。

My to do list.

Items

ID	登録日	タイトル	締切日
1	2013/05/29 11:01	Symfony2の勉強	ASAP

*** テンプレート(base.html)を継承 ***

My to do list.

Symfony2の勉強

Created: 2013/05/29 11:01

Updated: 2013/05/29 11:01

Symfony2の勉強を通してToDoListアプリを作ってみる。

Deadline is ASAP

[一覧に戻る](#)

*** テンプレート(base.html)を継承 ***

13. bootstrap を使って見栄えを変える

13.1 bootstrap をインストール

13.1.1 bootstrap を以下からダウンロード

<http://twitter.github.io/bootstrap/getting-started.html>

13.1.2 下記ディレクトリに展開

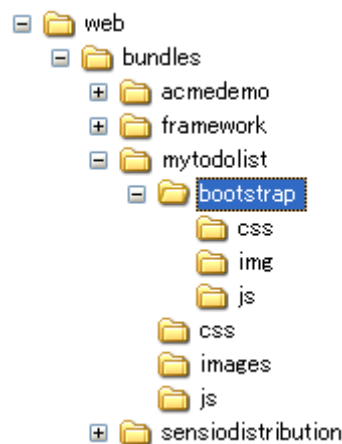
src/My/ToDoListBundle/Resources/public/

13.1.3 web ディレクトリ以下にシンボリックリンクを張る

上記の様に、エラーが出てしまう。指示に従って「--symlink」を使わずに symlink をする。

```
C:\xampp\htdocs\Symfony>php app/console assets:install web  
  
Installing assets using the hard copy option  
Installing assets for Symfony\Bundle\FrameworkBundle into  
web/bundles/framework  
Installing assets for My\ToDoListBundle into  
web/bundles/mytodolist  
Installing assets for Acme\DemoBundle into web/bundles/acmedemo  
Installing assets for Sensio\Bundle\DistributionBundle into  
web/bundles/sensiodistrib
```

これにより、bootstrap が web/bundles/mytodolist の中にハードコピーされる。



13.2 base.html.twig に bootstrap を設定

<head></head>の間に、bootstrap へのリンクを設定する。

```
~省略~  
<head>  
  <meta http-equiv="Content-Type" content="text/html;  
  charset=utf-8" />  
  
  {% block stylesheets %}  
    <link rel="stylesheet"  
href="{{ asset('bundles/mytodolist/bootstrap/css/bootstrap.m  
in.css') }}">  
  {% endblock stylesheets %}
```

```
<script type="text/javascript"
src="//ajax.googleapis.com/ajax/libs/jquery/1.8/jquery.min.js"></script>
~省略~
```

src=MyToDoListBundleResourcesviewbase.html.twig

13.3 各項目のスタイルを設定

ToDoList 画面

My to do list. on symfony2

Those items must be done within the deadlines...

ID	登録日	タイトル	締切日
1	2013/05/29 11:01	Symfony2の勉強	ASAP

*** テンプレート(base.html)を継承 ***

詳細表示画面

My to do list. on symfony2

Symfony2の勉強

#1	Symfony2の勉強を通してToDoListアプリを作って 一歩に近 みる。	Deadline is ASAP Created: 2013/05/29 11:01 Updated: 2013/05/29 11:01
----	--	--

*** テンプレート(base.html)を継承 ***

14. アイテムの新規追加機能追加

14.1 コントローラ追加

~省略~

```
use Symfony\Component\HttpFoundation\Request;
use MyToDoListBundle\Entity\Post;

class ToDoListController extends Controller
{
    ~省略~

    /**
     * @Route("/new", name="todo_new")
     */
    public function newAction(Request $request)
    {
        //フォームを作るメソッド createFormBuilder
        $form = $this->createFormBuilder(new Post())
            ->add('title')
            ->add('note')
            ->add('deadline')
            ->getForm();
    }
}
```

```
        if ('POST' === $request->getMethod()) {
            $form->bind($request);
            // バリデーション
            if ($form->isValid()) {
                $post = $form->getData();
                $post->setCreatedAt(new \DateTime());
                $post->setUpdatedAt(new \DateTime());
                $em = $this->getDoctrine()->getManager();
                $em->persist($post);
                $em->flush();

                return $this->redirect($this->generateUrl('todo_index'));
            }
        }

        return array(
            'form' => $form->createView(),
        );
    }
}
```

～省略～

src/My/ToDoListBundle/Controller/ToDoListController.php

14.2 index.html.twig に「New task!!!」ボタンを設置

```
～省略～

{% block content %}
<div class="span12">
    <div>
        <p>
            <a class="btn btn-primary" href="{{ path('todo_new') }}">
                New task!!!
            </a>
        </p>
    </div>
</div>
```

src/My/ToDoListBundle/Resources/views/ToDo/index.html.twig

14.3 new.html.twig を作成

```
{% extends 'MyToDoListBundle::base.html.twig' %}

{% block page_title 'Create new task!!!' %}

{% block content %}
<form action="{{ path('todo_new') }}" method="post">
    {{ form_enctype(form) }}
    {{ form_widget(form) }}
    <button class="btn btn-primary">new</button>
    <a href="{{ path('todo_index') }}">Return to the list.</a>
</form>

{% endblock content %}
```

src/My/ToDoListBundle/Resources/views/todo/new.html.twig

My to do list. on symfony2

Those items must be done within the deadlines...

New task!!			
ID	登録日	タイトル	締切日
1	2013/05/29 11:01	Symfony2の勉強	ASAP
2	2013/05/30 08:35	新規追加機能の追加	出来るだけ早く。

*** ダウンロード@base.htmlを継承 ***

体裁は後で整えよう...

15. 入力バリデーション追加

15.1 post.php にバリデーションを追加

～省略～

```
use Symfony\Component\Validator\Constraints as Assert;
```

```
class Post  
{
```

～省略～

```
/**  
 * @var string  
 *  
 * @ORM\Column(name="title", type="string", length=255)  
 * @Assert\NotBlank()  
 * @Assert\Length(min="2", max="50")  
 */  
private $title;
```

```
/**  
 * @var string  
 *  
 * @ORM\Column(name="note", type="text")  
 * @Assert\NotBlank()  
 * @Assert\Length(min="10")  
 */  
private $note;
```

```
/**  
 * @var string  
 *  
 * @ORM\Column(name="deadline", type="text")  
 * @Assert\NotBlank()  
 * @Assert\Length(min="10")  
 */  
private $deadline;
```

～省略～

src/My/ToDoListBundle/Entity/Post.php

15.2 バリデーションが効いているか確認

Title

- This value is too short. It should have 2 characters or more.

Note

- This value is too short. It should have 10 characters or more.

Deadline

- This value is too short. It should have 10 characters or more.

[new](#) [Return to the list.](#)

日本語にするには下記を変更すればよいとのことだが・・・

～省略～

```
#translator:      { fallback: %locale% }
```

～省略～

Symfony¥app¥config¥config.yml

16. 削除機能の追加

16.1 コントローラ追加

～省略～

```
class BlogController extends Controller
{
```

～省略～

```
/**
 * @Route("/{id}/delete", name="todo_delete")
 */
function deleteAction($id)
{
    $em = $this->getDoctrine()->getEntityManager();
    $post = $em->getRepository('MyToDoListBundle:Post')->find($id);
    if (!$post) {
        throw $this->createNotFoundException('The item does not exist');
    }
    // 削除
    $em->remove($post);
    $em->flush();

    return $this->redirect($this->generateUrl('todo_index'));
}
```

～省略～

src/My/ToDoListBundle/Controller/ToDoListController.php

16.2 テンプレート追加

～省略～

```
<tr>
  <td>{{ post.id }}</td>
  <td>{{ post.createdAt|date('Y/m/d H:i') }}</td>

  <td><a href="{{ path('todo_detail', {'id':
post.id}) }}">{{ post.title }}</a></td>

  <td>{{ post.deadline }}</td>
  <td><a class="btn btn-danger" href="{{ path('todo_delete',
{'id':post.id}) }}">delete</a></td>
</tr>
```

～省略～

src/My/ToDoListBundle/Resources/views/ToDo/index.html.twig

16.3 確認

http://localhost:8000/app_dev.php/blog/

My to do list. on symfony2

Those items must be done within the deadlines...

New task!				
ID	登録日	タイトル	締め切り	
1	2013/05/29 11:01	Symfony2の勉強	ASAP	delete
2	2013/05/30 08:35	新規追加機能の追加	出来るだけ早く。	delete

*** テンプレート (base.html) を継承 ***

17. 編集機能の追加

17.1 コントローラ追加

～省略～

```
class BlogController extends Controller
{
    /**
     * @Route("/new", name="todo_new")
     */
    public function newAction(Request $request)
    {
        $post = $post = new Post();
        // フォームの組立
        $form = $this->createFormBuilder($post)
```

～省略～

```
        return array(
            'post' => $post,
            'form' => $form->createView(),
        );
    }
}
```

～省略～

```
/**
 * @Route("/{id}/edit", name="todo_edit")
 */
public function editAction(Request $request, $id)
{
    // DB から取得
    $em = $this->getDoctrine()->getManager();
    $post =
    $em->getRepository('MyToDoListBundle:Post')->find($id);
    if (!$post) {
        throw $this->createNotFoundException('The item does
not exist');
    }
    // フォームの組立
    $form = $this->createFormBuilder($post)
        ->add('title')
        ->add('note')
        ->add('deadline')
        ->getForm();

    // バリデーション
    if ('POST' === $request->getMethod()) {
        $form->bind($request);
        if ($form->isValid()) {
            // 更新されたエンティティをデータベースに保存
            $post = $form->getData();
            $post->setUpdatedAt(new \DateTime());
            $em->flush();
            return
        }
        $this->redirect($this->generateUrl('todo_index'));
    }
    return array(
        'post' => $post,
        'form' => $form->createView(),
    );
}
```

src/My/ToDoListBundle/Controller/ToDoListController.php

17.2 編集用テンプレート追加

```
{% extends 'MyToDoListBundle:ToDoList:new.html.twig' %}
{% block page_title 'edit item' %}
```

src/My/ToDoListBundle/Resources/views/ToDo/edit.html.twig

17.3 新規作成用テンプレートを修正

修正前

～省略～

```
<form action="{{ path('todo_new') }}" method="post"
{{ form_enctype(form) }}>
    {{ form_widget(form) }}
    <button class="btn btn-primary">new</button>
    <a href="{{ path('todo_index') }}">Return to the list.</a>
</form>
```

～省略～

修正後

～省略～

```
<form action="{{ post.id ? path('todo_edit', {'id': post.id}) :
path('todo_new') }}" method="post" {{ form_ctype(form) }}>

    {{ form_widget(form) }}

    {% if post.id %}
        <button class="btn btn-info">Edit</button>
    {% else %}
        <button class="btn btn-primary">New{% endif %}</button>
</form>
```

～省略～

src¥My¥ToDoListBundle¥Resources¥views¥ToDo¥new.html.twig

17.4 編集ボタンを配置

～省略～

```
<tr>
    <td>{{ post.id }}</td>
    <td>{{ post.createdAt|date('Y/m/d H:i') }}</td>

    <td><a href="{{ path('todo_detail', {'id':
post.id}) }}">{{ post.title }}</a></td>

    <td>{{ post.deadline }}</td>
    <td><a class="btn btn-action" href="{{ path('todo_edit',
{'id':post.id}) }}">edit</a></td>
    <td><a class="btn btn-danger" href="{{ path('todo_delete',
{'id':post.id}) }}">delete</a></td>
</tr>
～省略～
```

src¥My¥ToDoListBundle¥Resources¥views¥ToDo¥index.html.twig

17.5 確認

http://localhost:8000/app_dev.php/todo/

My to do list. on symfony2

Those items must be done within the deadlines...

New task!!!			
ID	登録日	タイトル	締切日
5	2013/05/30 09:34	編集ボタン作成	今日中。
6	2013/05/30 09:34	待機を整える。	出来るだけ早くやる。

*** テンプレート(base.html)を継承 ***

18. 80 番ポートが何かに占有されている場合の対策

xampp の Control Panel を起動して、apache をスタートさせようとした時、他の何かに 80 番ポートが占有されていて、上手くスタートできない場合がある。その時は下記対策を試みる。

18.1 Visual Web Developer Express のサービスを停止

Visual Web Developer Express と同時にインストールしたと思われる WebMatrix は Windows 起動時に 80 番ポートを占有する。よってサービスを停止する。

スタートメニューの「コンピュータ」を右クリック→「管理」→「サービスとアプリケーション」→「サービス」から、「Web Deployment Agent Service」を停止させる。

18.2 Skype のポート変更、または自動起動なくする

18.2.1 自動起動をやめる（お勧め）

Windows 起動時に、自動的に Skype が起動して 80 番ポートを占有してしまう可能性が有る為、Skype は必要な時に起動するよう設定する。

Skype のメニューバーで「ツール」メニュー > 「設定…」をクリック。
左側のメニューから「一般設定」を選択し、右側の「コンピュータ起動時-自動的に Skype を起動」のチェックを外す。
「保存」をクリックする。

18.2.2 ポート変更

80 番ポート以外を設定する。ただし別のアプリケーション（Apache HTTP サーバや IIS など）がこの 80 番ポートを使用している場合、Skype が正常に機能しない可能性あり。

Skype のメニューバーで「ツール」> 「設定…」をクリック。
「詳細」で、「接続」をクリック。
Skype の着信接続に使用するポートを入力。
「保存」をクリックします。

19. `php -r "eval('=...');"</codeとは`

composer.phar をインストールする際、ターミナルから下記を実行した。

```
php -r  
"eval('?'>'.file_get_contents('https://getcomposer.org/installer'))';"
```

要素それぞれの意味は・・・

php -r : <?...?>タグなしで PHP のコードを実行することが出来るオプション。

eval : 文字列を PHP コードとして評価する。つまり自動的に<?...?>で囲んだ状態にする。

file_get_contents : 引き数（ファイルや html）の中身をゴツゴツ取得する PHP 関数。

なぜこのような事をやっているのか・・・

そもそも PHP はサーバサイドスクリプト、Web サーバ上で動かすことで HTML 上に動的な要素を作ることが出来るというもの。

詳しくは分からないが、'https://getcomposer.org/installer'の中に記述された PHP の処理は、クライアントで動かすことで、クライアントに composer.phar がインストールされるように記述されたものであって、そのままクライアントに持ってきて PHP として実行しても上手く動くのだと思う。

そうしない代わりに、'https://getcomposer.org/installer'の中身をゴツゴツ読み込んだ際、「eval('?'>）」で受けることで PHP タグをないものとして読込、更に「php -r」に渡すことで PHP として実行しているのだと思う。