

SYMPHONY 1.2

BRENT SHAFFER
TRAVIS BLACK

SYMFONY OVERVIEW

- object oriented
- MVC
- abstractable libraries
- environment configurable
- cascading configuration
- test framework
- form framework
- auto caching
- i18n support
- RESTful routing
- PAKE tasks
- CRUD / admin generator
- easily extendible
- active community

GETTING STARTED

Download and install symfony

- Sandbox
 - http://www.symfony-project.org/tutorial/1_2/my-first-project
- \$./symfony generate:project projectname
- \$./symfony generate:app frontend

GETTING STARTED

- \$./symfony freeze ~/Sites/symfony/1.2/data/
- edit config / ProjectConfiguration.class.php

```
$this->enableAllPluginsExcept(array('sfPropelPlugin', 'sfCompat10Plugin'));
```

- edit config / databases.yml

```
all:
  doctrine:
    class:      sfDoctrineDatabase
    param:
      classname: DoctrinePDO
      dsn:       mysql:dbname=topos;host=localhost
```

CREATE YOUR BLOG MODEL

- edit your schema.yml

```
BlogPost:
  columns:
    title:
      type: string(255)
    description:
      type: clob
      notnull: true
    pubdate:
      type: timestamp
    author:
      type: string(255)
```

- Build your database, models, and forms
 - `$./symfony doctrine:build-all-reload`
- Take a second to examine generated classes

CREATE YOUR BLOG MODULE

- `$./symfony generate:module frontend blog_post`
- By default, pages are viewable at `/:module/:action`
- browse to http://localhost/myproject/blog_post
- Default action is *index*

ADD YOUR DATA FIXTURES

- create a `data/fixtures/BlogPost.yml`

```
BlogPost:
  my_blog_post:
    title:      First Post
    description: |
      <p>
        ante. egestas. augue, Vestibulum lacus Nam senectus sed,
        sagittis sit Mauris erat commodo tincidunt dui vitae,
        egestas Aenean Donec malesuada tincidunt eleifend tortor
        fames metus accumsan quam condimentum rutrum eros pulvinar
        Pellentesque et egestas Donec neque Ut amet leo. pharetra.
        sit ullamcorper est placerat et non quam, eu dapibus,
        tempus orci, fermentum, in amet, facilisis. semper. dui.
        netus ac tempor erat enim vitae faucibus, mi, vulputate
        elit amet sit ornare eget erat. eros turpis ipsum wisi,
        ultricies Praesent mi cursus habitant felis. tortor
      </p>
    timestamp:  "2009-30-31"
    author:      Brent Shaffer
```

- Now we can manipulate our data
 - `./symfony doctrine:dql "FROM BlogPost"`

EDIT YOUR ACTIONS

- Pull in your Blog objects

```
public function executeIndex(sfWebRequest $request)
{
    $q = Doctrine_Query::create()
        ->from('BlogPost b');

    $this->posts = $q->execute();
}
```

- Edit your View (*templates/indexSuccess.php*)

EXAMINE YOUR LAYOUT

- Layout is rendered after modules
- `$sf_content`, `$sf_request`, `$sf_user`,
`$sf_response`, `$sf_request` available

ADMIN GENERATOR

- `$./symfony doctrine:generate-admin backend BlogPost`
- Examine `generator.yml`, Form Classes
- **OUT-OF-THE-BOX AUTHENTICATION**
- **FORM FRAMEWORK**

DOCTRINE BEHAVIORS

- Install csActAsCommentablePlugin
 - `$ svn co path/to/plugin/csActAsCommentablePlugin plugins/csDoctrineActAsCommentablePlugin`
- Add behavior to schema.yml

```
BlogPost:  
  actAs:  
    Commentable:
```

DOCTRINE BEHAVIORS

- Rebuild your models, clear your cache
- Add the component to your view

```
<?php use_helper('Comment') ?>  
<div class='comments'>  
    <?php include_component('csComments', 'thread', array('record' => $post)) ?>  
</div>
```

ADD A RELATION

- edit your schema.yml

```
BlogPost:
  columns:
    title:
      type: string(255)
    description:
      type: clob
      notnull: true
    pubdate:
      type: timestamp
    blog_author_id:
      type: integer
  relations:
    BlogAuthor:
      foreignAlias: Posts

BlogAuthor:
  columns:
    first_name:
      type: string(255)
    last_name:
      type: string(255)
```