Blast Project

Release

Nov 03, 2017

Reference Guide

CHAPTER 1

Installation

Blast CoreBundle can be installed at any moment during a project's lifecycle, whether it's a clean Symfony installation or an existing project.

1.1 Prerequisites

- having a working Symfony2 environment
- having created a working Symfony2 app (including your DB and your DB link)
- having composer installed (here in /usr/local/bin/composer, with /usr/local/bin in the path)

1.2 Downloading

```
$ composer require blast-project/core-bundle dev-master
```

This will download and install :

- knplabs/knp-menu
- knplabs/knp-menu-bundle
- cocur/slugify
- sonata-project/core-bundle
- sonata-project/cache
- sonata-project/block-bundle
- sonata-project/exporter
- twig/extensions
- sonata-project/admin-bundle

- sonata-project/doctrine-orm-admin-bundle
- blast-project/core-bundle
- libre-informatique/base-entities-bundle
- twig/twig ^1.22.1

1.3 Third party bundles, Sonata bundles

Please refer to the Sonata Project's instructions, foundable here :
https://sonata-project.org/bundles/admin/3-x/doc/reference/installation.html
https://sonata-project.org/bundles/user/3-x/doc/reference/installation.html

And follow the installation guides.

At the end, you should have a `app/AppKernel.php` that looks like that:

```
use Symfony\\Component\\HttpKernel\\Kernel;
use Symfony\\Component\\Config\\Loader\\LoaderInterface;
```

CHAPTER 2

Getting started with Blast CoreBundle

2.1 Introduction

The goal of this bundle is to make the use of SonataAdmin "view-models" possible without writting a line of PHP, without loosing a feature of Sonata, and importing the idea of composite settings using lots of characteristics of an admin (its direct inheritance tree, the traits used by its Entity, the inheritance tree of its Entity…), making things more flexible, extendable, reusable and maintenable through many bundles and uses.

This bundle is the next step after the SonataAdminBundle. Configure an entire backend bundle filling only YAML files… Try it!

It is also the core of Libre Informatique's Symfony 2/3 projects.

2.2 Example

I want to design and create a bundle as a toolbox for other bundles' entities. It will provide traits for email addresses and phonenumbers, for instance (cf. BlastBaseEntitiesBundle).

Using the BlastCoreBundle, your "base" bundle will carry the traits, but also the way to display properties given by its traits in a SonataAdmin (which becomes a CoreAdmin) CRUD. Then using the traits of your "base" bundle in the entities of other bundles (also implementing the BlastCoreBundle) will add the fields naturally, the columns in the list of objects, etc… as you set up for your trait in your "base" bundle, without having to write a line for this.

Imagine this feature appliable to 50 entities distributed in 10 bundles, and count in your mind the number of saved lines, the number of potential bugs avoided and the ease of maintenance when you want to change the nature of the field used by the provided email address or phonenumber… This is what the BlastCoreBundle permits.

CHAPTER 3

Configuration

CHAPTER 4

Architecture

CHAPTER 5

The Contribution Guide

Note:

This section is based on the great Symfony documentation.
The following is a set of guidelines for contributing to Blast on GitHub.

5.1 How to install Blast to contribute?

Before you start contributing you need to have your own local environment for editing things.

To install Blast main application from our main repository and contribute, run the following command:

```
$ composer create-project -s dev blast-project/blast
```

5.2 Reporting bugs and suggesting enhancements

Before creating issues, please check this list as you might find out that you don't need to create one. When you are creating a new issue, please include as many details as possible to help maintainers reproduce the problem or understand your suggestion.

5.3 Submitting changes

Like most projects, we propose a standard GitHub Flow for contributions:

1. Fork
2. Create a topic branch

3. Add commits
4. Create a Pull Request
5. Discuss and review your code
6. Merge

If you want to submit changes, please send a GitHub Pull Request with a clear list of what you've done (read more about pull requests).

Please make sure all of your commits are atomic (one feature per commit) and always write a clear log message for your commits to help maintainers understand and review your submission.

CHAPTER 6

Other Bundles

- BaseEntitiesBundle: provides some tools for a better integration in Sonata Admin