При реализации требуется придерживаться стандартов, упомянутых в документе. Результат присылать в виде ссылки на github-репозиторий.

Базовые материалы по Symfony и сопутствующим технологиям

1. PHP. The Right Way [http://www.phptherightway.com](http://www.phptherightway.com/)

2. Composer <https://getcomposer.org/doc/00-intro.md>

3. Symfony

* Фреймворк <http://symfony.com/doc/current/book/index.html> (Все главы кроме HTTP Cache, Databases and Propel)
* Компоненты Symfony (Console, DependencyInjection, EventDispatcher, Finder, Process, YAML, Security)
* Много полезных материалов в Cookbook <http://symfony.com/doc/current/cookbook/index.html> (Рекомендуется регулярно обращаться в этот раздел, много полезной информации)

4. Стандарты кода

* <http://symfony.com/doc/current/contributing/code/standards.html> (а также PSR-0, PSR-1, PSR-2, которые упомянуты в статье)
* <http://symfony.com/doc/current/contributing/code/conventions.html>
* <http://twig.sensiolabs.org/doc/coding_standards.html>
* В своих проектах, в том числе в текущем настраивать в локальном репозитории <https://github.com/intaro/php-git-hooks>

Необходимо сделать следующий проект «Личная библиотека книг».

**Личная библиотека книг**

1. Сделать в проекте авторизацию (на основе FOSUserBundle)

В app/console есть команды для создания пользователя.

2. Главная страница

*Неавторизованный режим*

Выводим список книг в порядке прочтения с указанием названия, автора и обложки, а также ссылка на скачивание, если она разрешена для книги. Предусмотреть кеширование запроса получения списка книг на 24 часа. При редактировании/добавлении книги кеш должен сбрасываться.

*Авторизованный режим*

У каждой книги кроме описанного выше выводится ссылка Редактировать. Также в начале страницы выводится ссылка Добавить книгу.

3. Страница Добавить книгу

Выводится форма с полями:

* Название
* Автор
* Обложка (png, jpg)
* Файл с книгой (до 5мб)
* Дата прочтения
* Галочка «Разрешить скачивание»

Обложку и файл с книгой сохранять таким образом, чтобы не было конфликтов одинаковых названий и проблемы скопления большого количества файлов в одной папке.

4. Редактировать книгу

Аналогично созданию, только в форме отображаются текущие данные по книге, которые можно изменить. Файлы обложки и книги в режиме редактирования можно удалить.

5. Подготовить Doctrine Subscriber, которые будет срабатывать на удаление книги и удалять файлы обложки и самой книги

6. Сделать extenstion к twig, в котором объявить функцию для вывода картинок в заданном размере

7. Сделать 3 API-метода: api/v1/books, api/v1/books/{id}/edit и /api/v1/books/add, которые будут позволять соответственно получать список книг, редактировать существующую книгу и добавлять новую книгу. Первый метод возвращает всю информацию, ссылки на файл книги и обложку абсолютные. Если скачивание не разрешено, то файл не выдается. В редактировании и добавлении можно работать со всеми полями кроме файла и обложки.

Предлагается использовать JMS/Serializer при реализации. Доступ к API по apiKey, который задается в параметрах. Ключ передается в GET/POST-параметре.

8. В качестве СУБД использовать PostgreSQL

9. Проект разместить в репозитории на github