

## Введение

Вероятнее всего, каждый из нас сталкивался с результатами работы различных генераторов документации. Общий принцип их работы следующий: на вход такого генератора поступает специальным образом комментированный исходный код, а иногда и другие компоненты программы, а на выходе создаётся готовая документация для распространения и использования.

Программы для генерации документации, позволяют нам получать документацию, предназначенную для программистов (документация на API) и/или для конечных пользователей системы, по особым образом комментированному исходному коду и, в некоторых случаях, по исполняемым модулям (полученным на выходе компилятора).

Обычно генератор анализирует исходный код программы, выделяя синтаксические конструкции, соответствующие значимым объектам программы (типам, классам и их членам/свойствам/методам, процедурам/функциям и т. п.). В ходе анализа также используется метаинформация об объектах программы, представленная в виде документирующих комментариев. На основе всей собранной информации формируется готовая документация, как правило, в одном из общепринятых форматов — HTML, HTMLHelp, PDF, RTF и других.

## Результат исследований

Список программ которые можно использовать для документации, весьма обширен, в данном случае мы рассмотрим, несколько которые я выделил из общего списка.

### Doxygen ([Ссылка на сайт](#))

**Doxygen** — кросс-платформенная система документирования исходных текстов, которая имеет широкий набор целевых языков: C++, C, Objective-C, Python, Java, C#, PHP, IDL, Fortran, VHDL и частично поддерживает D. Doxygen генерирует документацию на основе исходных текстов с комментариями особого вида и может быть настроен на извлечение структуры программы с недокументированными исходниками. Выходными форматами могут быть HTML, LATEX, man, rtf, xml. Doxygen используется в проектах KDE, Mozilla, Drupal, Pidgin, AbiWorld, FOX toolkit, Torque Game Engine, Crystal Space.

Перейдем к процессу установки и использованию данной программы.

Установка через терминал (один из вариантов установки, более простой в использовании):

Информация была взята с прсоторов интернета и коментариев опытных пользователей.

Установим сам **doxygen**:

```
sudo apt-get install doxygen
```

Установим **graphviz** для диаграмм:

```
sudo apt-get install graphviz
```

Установка графического отображения:

```
sudo apt-get install doxygen-gui
```

Теперь мы можем запустить doxygen в терминале:

```
doxywizard
```

Мы получаем такое окно настроек:

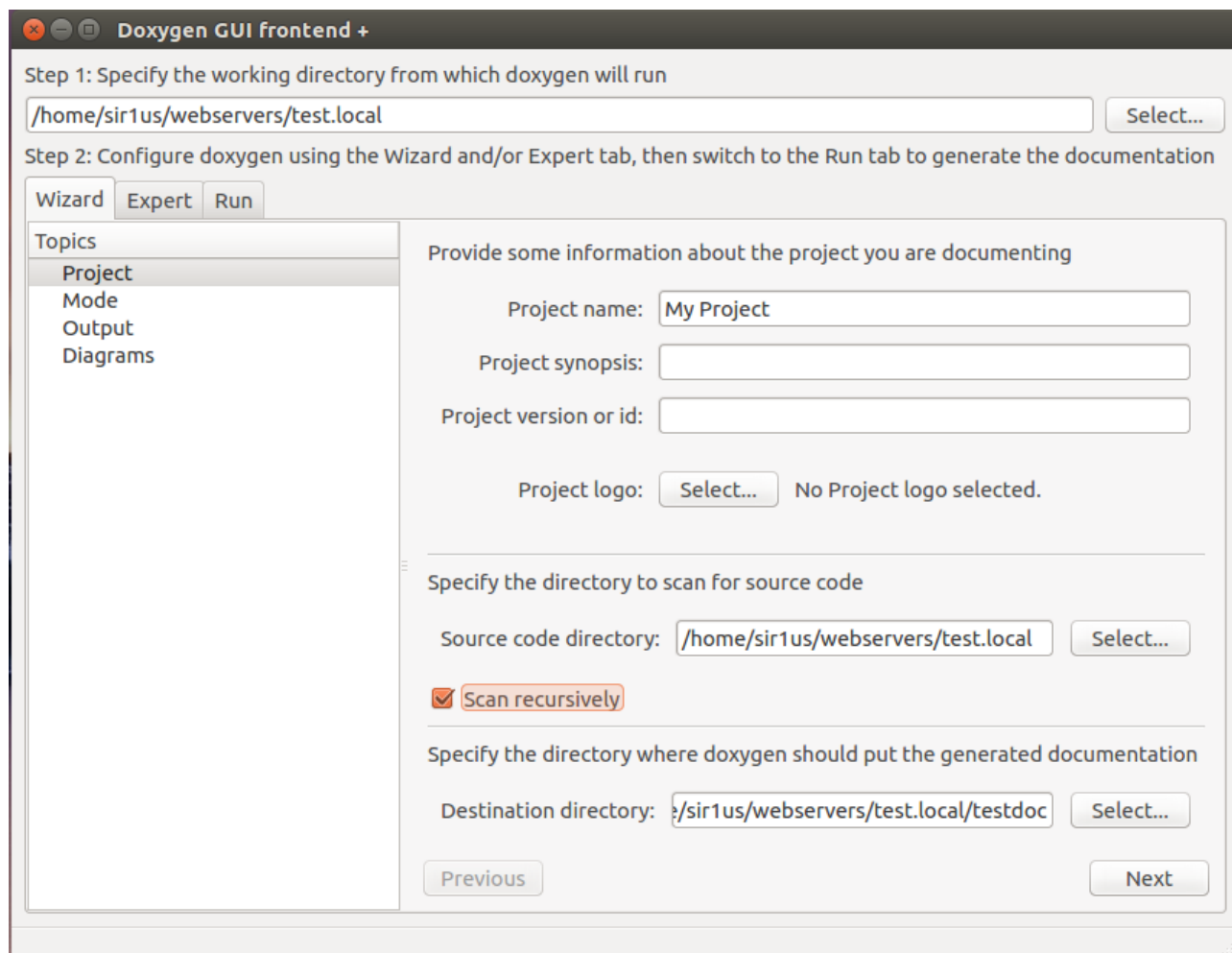
**Первый шаг:** Указываем директорию для документации.

В моем случае `/home/sir1us/webserver/phpacademy.local/rock/`

**Второй шаг:** Мы можем указать информацию о проекте и даже загрузить логотип, если надо. Данные поля не обязательны для заполнения.

Чуть ниже информация о **specify the directory to scan for source code** (укажите каталог для сканирования кода). Можем указать «./»

И в третьем поле с «**Select**» мы можем указать директорию куда будет помещена документация. Пример: новая папка - «./test» в этой директории



Теперь поговорим об остальных вкладках и полях.

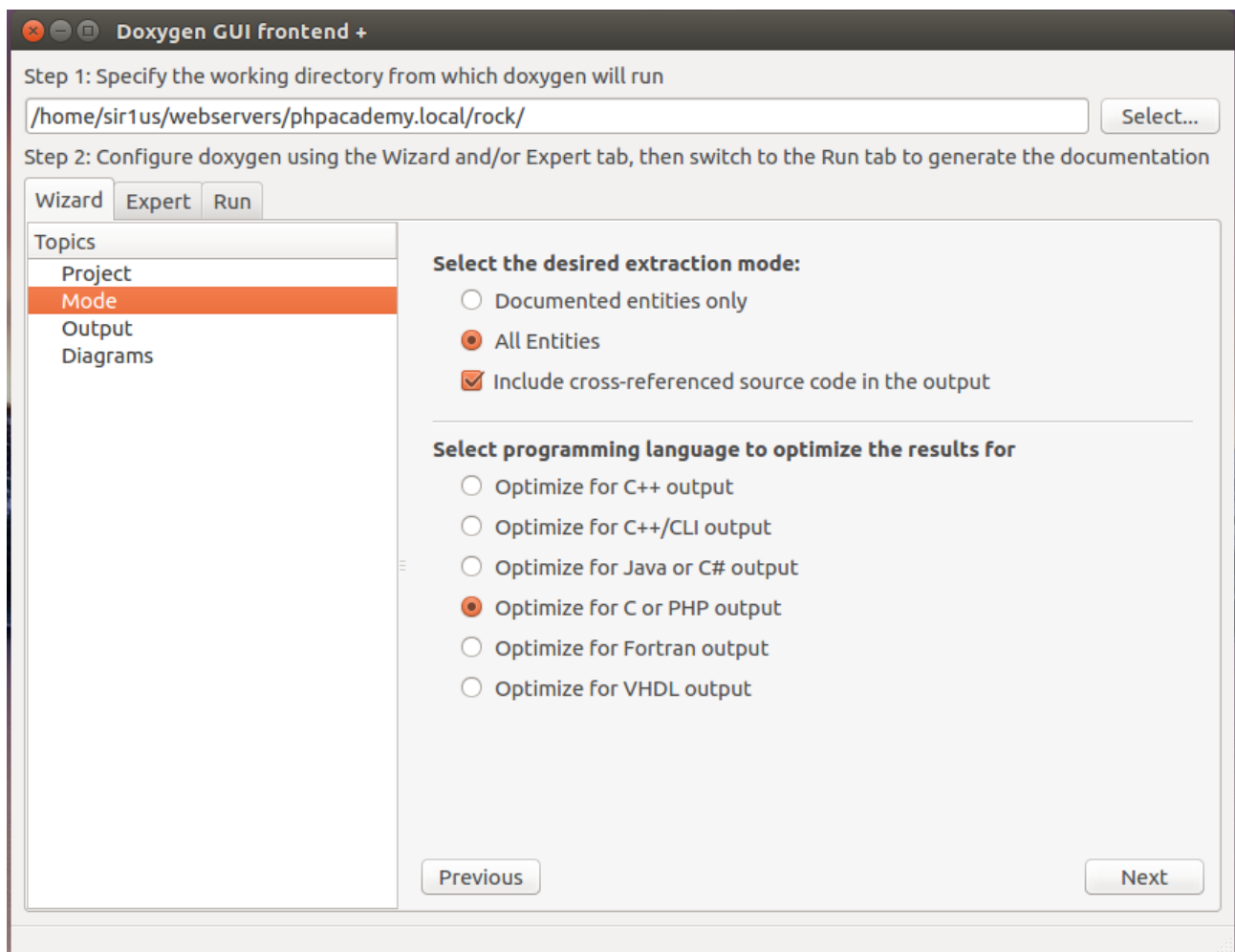
## Mode

С самого начала мы находимся во вкладке **Wizard**, строке настроек **Project** — ее мы уже настроили. Далее переходим ко второй строке **Mode**.

Нас просят выбрать «Режим извлечения». Рекомендую выбрать **All Entities** (Все объекты).

И так же поставить галочку в поле - **Include cross-referenced source code in the output** (включать ссылки кода в исходном варианте).

В пункте выбора языка программирования ставим галочку напротив **PHP** (Оптимизировать для C или PHP).

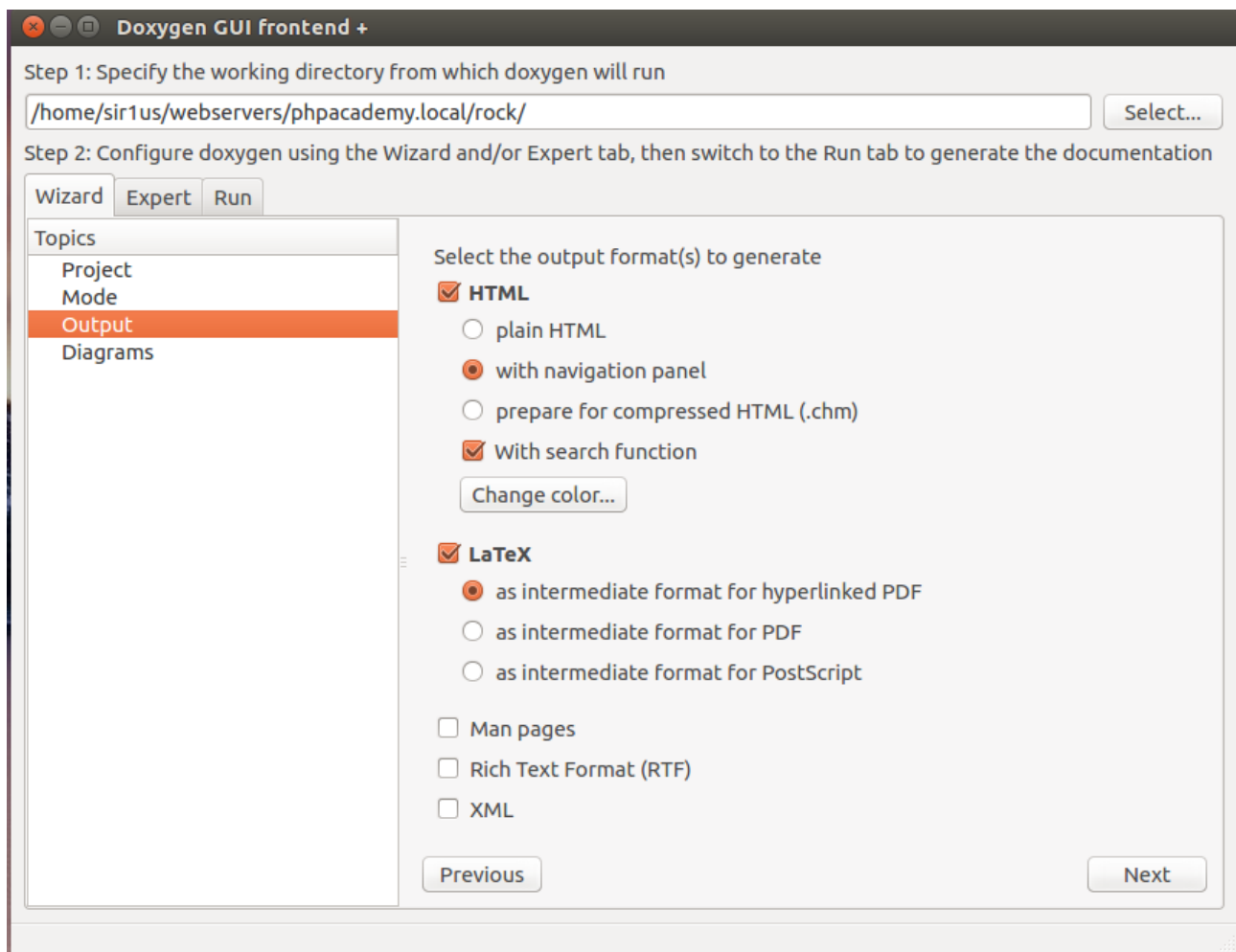


## OutPut

В данной строке мы настраиваем те папки которые мы получим, их отображение и некоторые дополнения.

По стандарту у нас будет две папки HTML и LATEX. В полях сразу выбраны стандартные варианты загрузки.

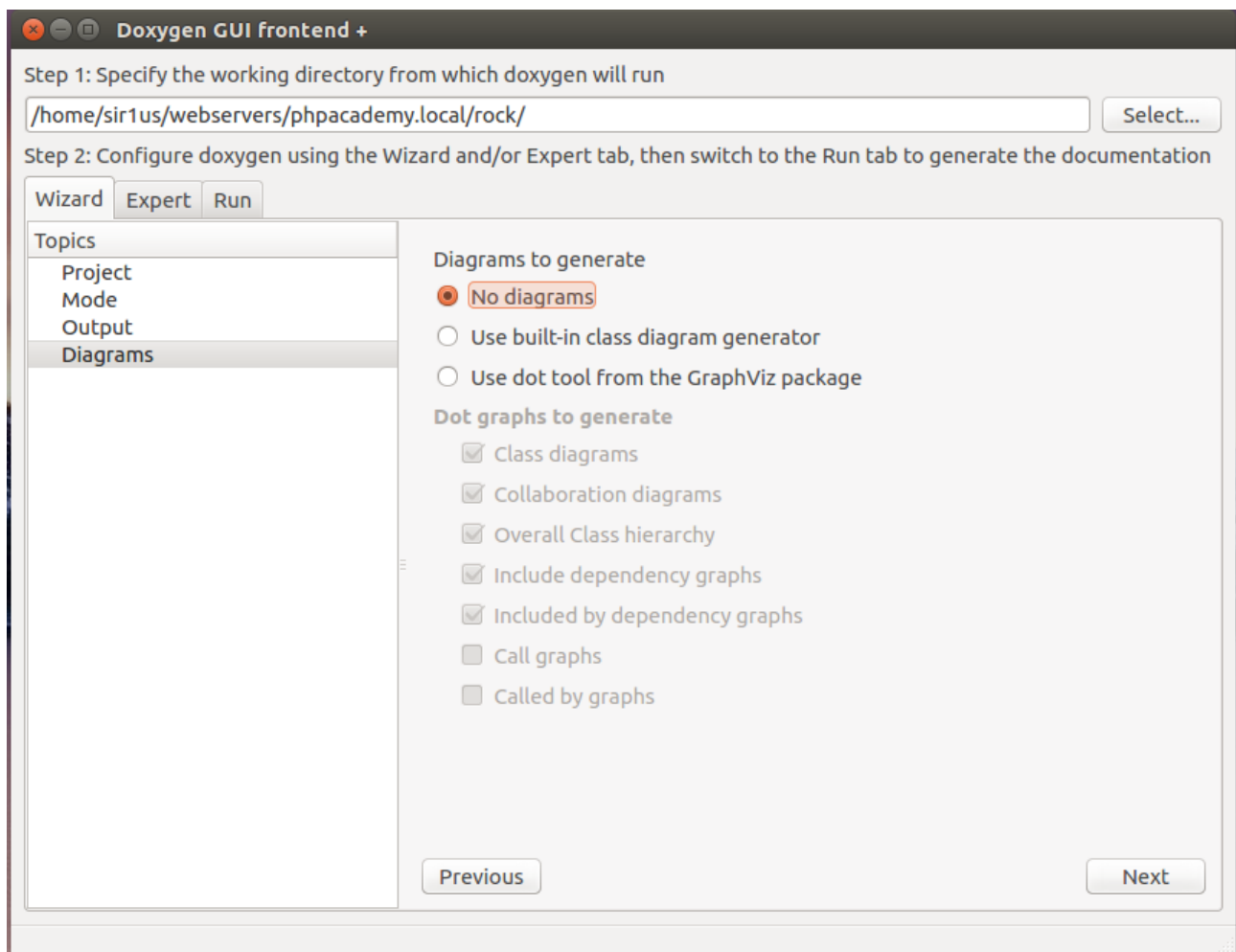
Рекомендую выбрать в поле HTML второй пункт — **with navigation panel**. Будет добавлена панель навигации для удобства.



## Diagrams

В этой строке можно настроить графическое отображение связей файлов в проекте.

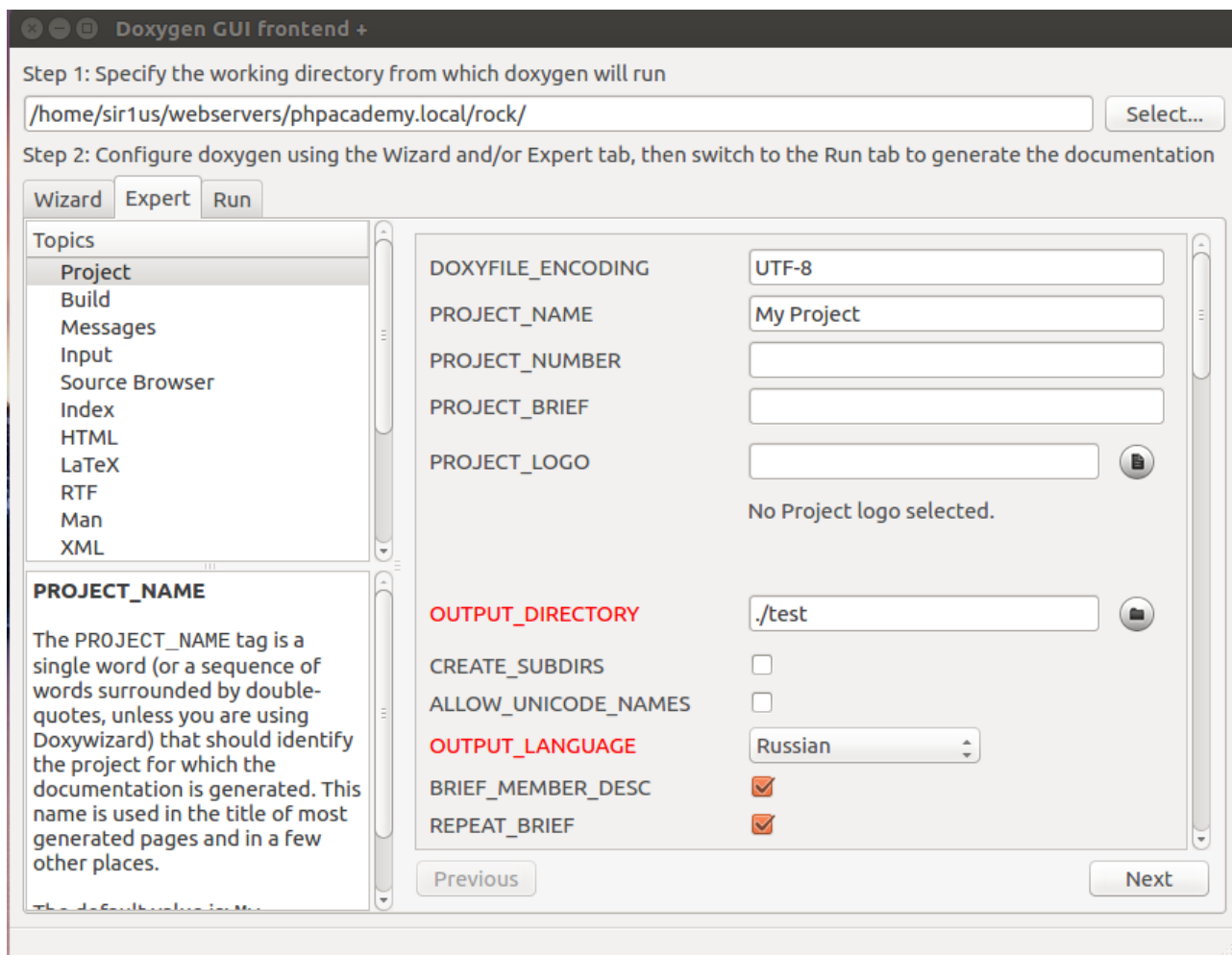
Так как, мне необходима только документация и мой пример это всего лишь несколько функций, то я пропущу данный пункт и просто выключу настройку диаграммы.



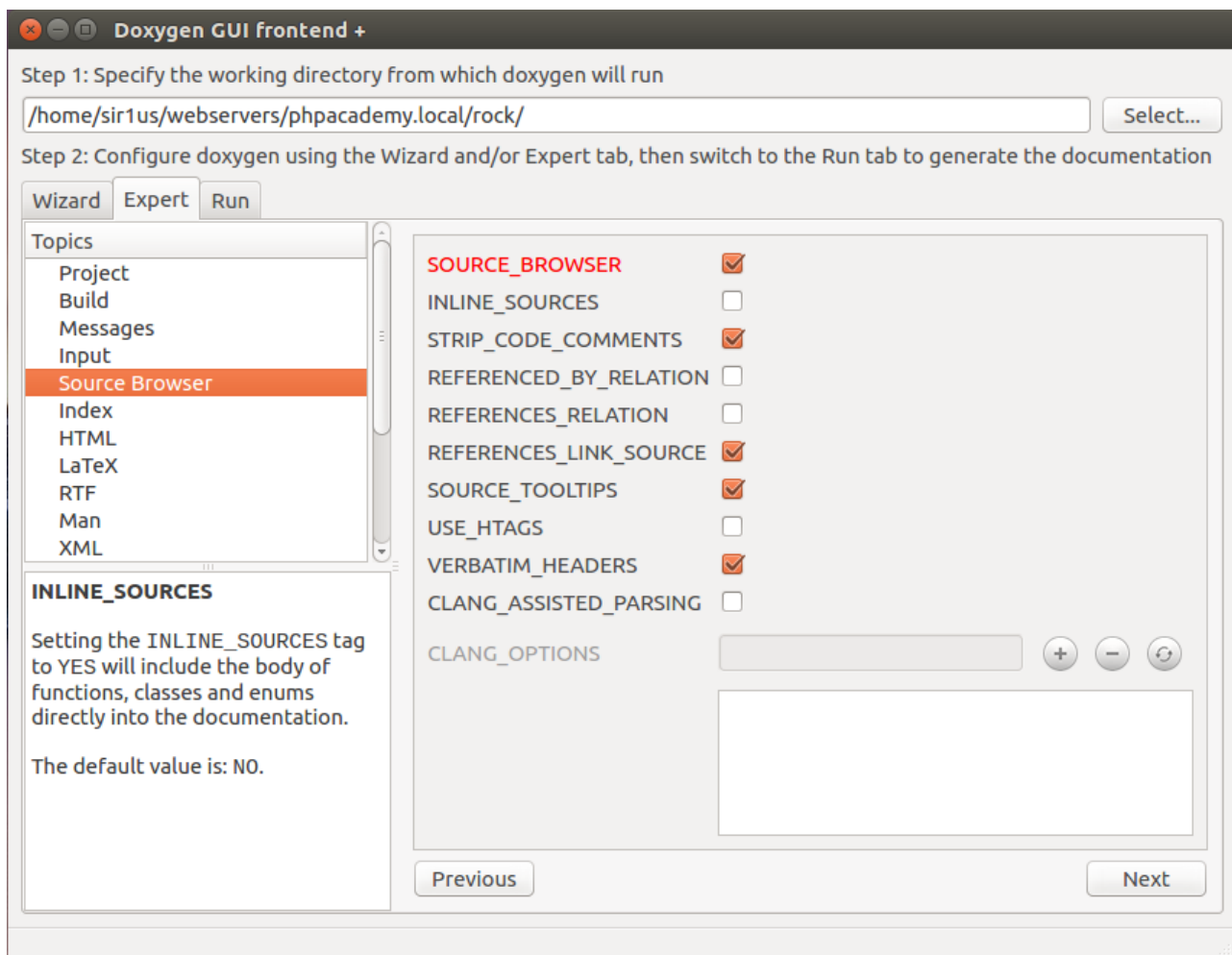
### Вкладка **Expert**

Подробные настройки. Для обычной выгрузки документации нам тут практически нечего не надо менять, но все же стоит уделить внимание нескольким моментам.

В строке **Project** рекомендую выбрать язык Русский либо другой удобный.



В строке **Source Browser** следует поставить галочку в пункте **SOURCE\_BROWSER**. Таким образом мы получим в документации ссылки на данные в коде.

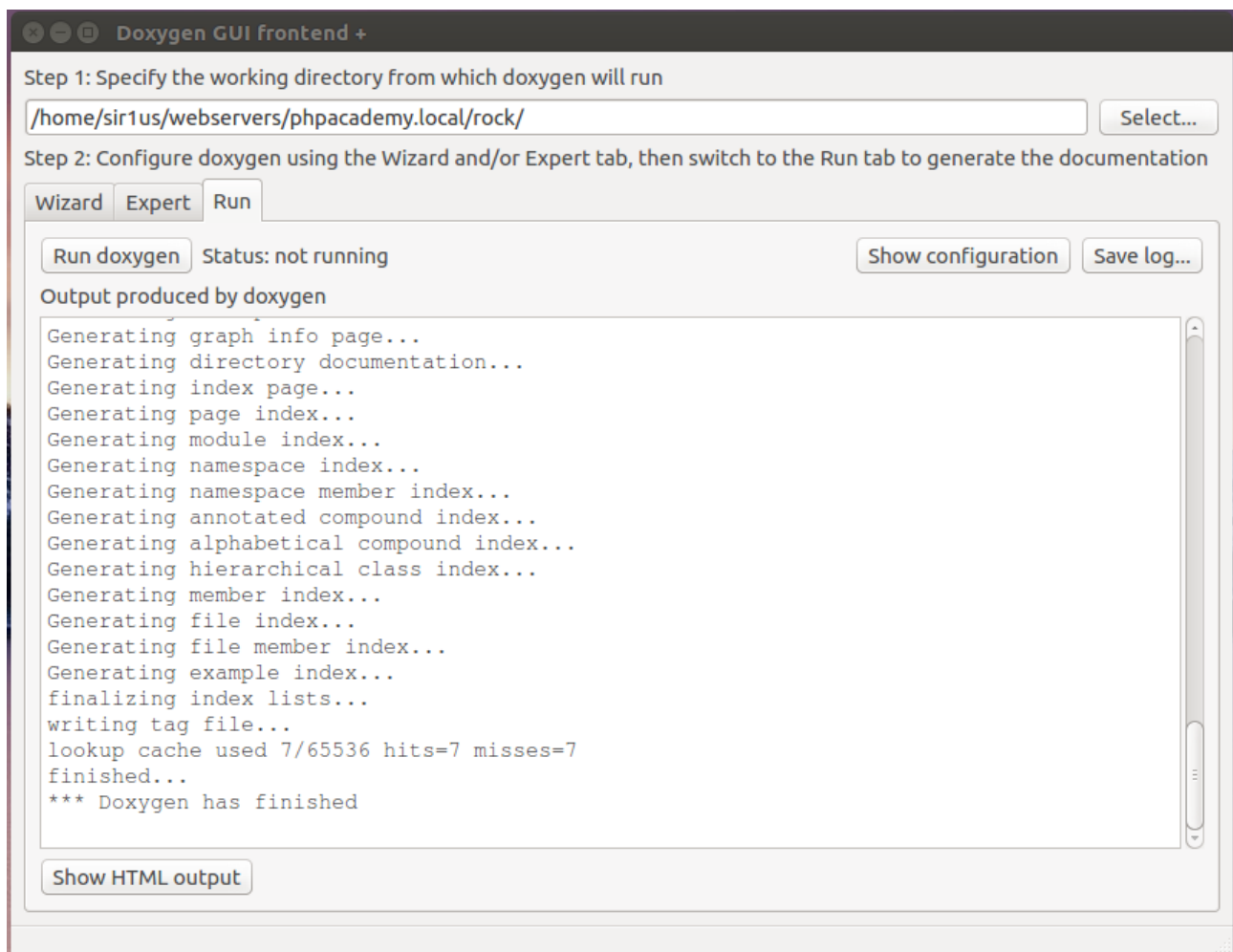


Более с подробными настройками рекомендую ознакомиться в мануале.

Заключительная вкладка **Run**

Запускаем Дохуген и получаем результат в выбранную ранее директорию.





Результат находится в папке Rock.

## PHPDoc или phpDocumentor ([Ссылка на сайт](#))

**phpDocumentor** — это система документирования исходных текстов на PHP. Имеет встроенную поддержку генерации документации в формате HTML, LaTeX, man, RTF и XML. Также вывод может быть легко сконвертирован в CHM, PostScript, PDF. Альтернативой использованию phpDocumentor является Doxygen.

### Процесс Установки и использования

На официальном сайте представлен вариант установки через репозиторий. Данный вариант выглядит просто и я думаю он весьма рабочий, но он поддерживает только 5 версию PHP, так как у меня уже была последняя версия PHP7 то возникли проблемы при установке. В PHP7 отсутствуют файлы в `php.ini` которые нужны для его работы.

Единственный вариант который помог решить данную проблему это запустить программу через `composer` взяв `dev` ветку.

```
composer require --dev phpdocumentor/phpdocumentor
```

Рекомендую сразу устанавливать файл в директорию которую документировать.

После установки мы получили несколько файлов.

Файлы самого `composer` и папку `vendor` в этой папке содержатся все что нам необходимо.

Для того что бы запустить PHPDoc нам достаточно одной команды в которой мы должны указать все маршруты.

Пример:

```
./vendor/bin/phpdoc -d path/to/my/project -f path/to/an/additional/file -t path/to/my/output/folder
```

`./vendor/bin/phpdoc` - расположение `.bat` файла.

`-f` — маршрут к директории или файлу для документации

`-d` — директория где расположены файлы для документации

`-t` — директория в которую сохраниться результат документации, если она отсутствует то PHPDoc создаст ее сам.

`--template="zend"` — данной командой можно выбрать шаблон для отображения, я использовал данный вывод.

`--title="Test"` — можно указать название документации.

Это основные теги для выполнения запроса, с более детальной информацией рекомендую ознакомиться в [мануале](#).

```

sirius@sirius-pc:~/webserver/phpacademy.local/rock$ php vendor/bin/phpdoc run -d ./phpdoc/ -t ./phpdoc/docs/
Collecting files .. OK
Initializing parser .. OK
Parsing files
Parsing /home/sirius/webserver/phpacademy.local/rock/phpdoc/rock-bands.php
No summary was found for this file
Storing cache in "/home/sirius/webserver/phpacademy.local/rock/phpdoc/docs" .. OK
Load cache .. 0.000s
Preparing template "clean" .. 0.007s
Preparing 17 transformations .. 0.000s
Build "elements" index .. 0.000s
Replace textual FQCNs with object aliases .. 0.000s
Resolve @link and @see tags in descriptions .. 0.000s
Enriches inline example tags with their sources .. 0.000s
Build "packages" index .. 0.000s
Build "namespaces" index and add namespaces to "elements" .. 0.000s
Collect all markers embedded in tags .. 0.000s
Transform analyzed project into artifacts ..
Applying 17 transformations
Initialize writer "phpDocumentor\Plugin\Core\Transformer\Writer\FileIo"
Initialize writer "phpDocumentor\Plugin\Twig\Writer\Twig"
Initialize writer "phpDocumentor\Plugin\Graphs\Writer\Graph"
Execute transformation using writer "FileIo"
Execute transformation using writer "FileIo"
Execute transformation using writer "FileIo"
Execute transformation using writer "FileIo"
Execute transformation using writer "FileIo"
Execute transformation using writer "twig"
Execute transformation using writer "twig"
Execute transformation using writer "twig"
Execute transformation using writer "twig"
Execute transformation using writer "twig"
Execute transformation using writer "twig"
Execute transformation using writer "twig"
Execute transformation using writer "twig"
Execute transformation using writer "twig"
Execute transformation using writer "twig"
Execute transformation using writer "twig"
Execute transformation using writer "Graph"
0.040s
Analyze results and write report to log .. 0.000s

```

В данном примере, моя строка запроса немного отличается, но она так же работает как и вариант в примере.

Для отображения дополнительной информации следует пользоваться тегами данной программы при документации самого кода.

## Tags

К примеру в описания кода мы можем оставить тег `@author`:

```

/**
 * функция, которая сортирует этот массив по кол-ву участников (band->members).
 * @param $array
 * @return mixed
 * @author Vladimir
 */

```

Который в сгенерированной документации выводит дополнительную строку с автором в разделе Теги.

Тегов достаточно много, по этому ознакомиться с ними можно в разделе [документации](#) на официальном сайте.

Результат выполнения так же есть в директории.

## ApiGen ([на Git](#), [сайт приложения](#))

**ApiGen** — быстрая и удобная программа для генерации документации. Она очень похожа по работе с PhpDocumentor, но выглядит более уверено по сравнению с ней. Разработчиками данной программы являются [IDE NetBeans](#), по этому ApiGen можно легко использовать с NetBeans

### Установка и функциональность

Установить ApiGen можно несколькими способами, через [PEAR](#) (как и PhpDocumentor) так и через composer или просто скачать с сайта файл [.phar](#) для установки. ([описание установки и использования с Git](#))

Установим ApiGen через composer:

```
composer require --dev apigen/apigen
```

Или

```
composer global require --dev apigen/apigen
```

После установки в нужную директорию у нас появиться папка [vendor](#). Для простого запуска генерации документации достаточно одной команды

```
./vendor/bin/apigen generate -s ./src -d ./docs
```

**-s** — маршрут к дериктории.

**-d** — дериктория в которую будут выгружены файлы

```
sirius@sirius-pc:~/webservers/phpacademy.local/rock$ ./vendor/bin/apigen generate -s ./rock-bands.php -d ./docs
Scanning sources and parsing
Found 0 classes, 0 constants and 6 functions
Generating API documentation
100 % - Finished!
sirius@sirius-pc:~/webservers/phpacademy.local/rock$
```

Для более детальной информации по генерированию можно написать команду **—help** которая предоставит информацию по разным возможностям программы.

```

sirius@sirius-pc:~/webserver/phpacademy.local/rock$ ./vendor/bin/apigen generate --help
Usage:
  generate [options]

Options:
  -s, --source=SOURCE           Dirs or files documentation is generated for. (multiple values allowed)
  -d, --destination=DESTINATION Target dir for documentation.
  --access-levels=ACCESS-LEVELS Access levels of included method and properties. [default: ["public","protected"]] (multiple values allowed)
  --annotation-groups=ANNOTATION-GROUPS Generate page with elements with specific annotation.
  --base-url=BASE-URL           Base url used for sitemap (useful for public doc).
  --config=CONFIG               Custom path to apigen.neon config file. [default: "/home/sirius/webserver/phpacademy.local/rock/apigen.neon"]
  --google-cse-id=GOOGLE-CSE-ID Custom google search engine id (for search box).
  --google-analytics=GOOGLE-ANALYTICS Google Analytics tracking code.
  --debug                       Turn on debug mode.
  --deprecated                  Generate documentation for elements marked as @deprecated
  --download                    Add link to ZIP archive of documentation.
  --extensions=EXTENSIONS       Scanned file extensions. [default: ["php"]] (multiple values allowed)
  --exclude=EXCLUDE             Directories and files matching this mask will not be parsed (e.g. */tests/*). (multiple values allowed)
  --groups=GROUPS               The way elements are grouped in menu. [default: "auto"]
  --charset=CHARSET             Charset of scanned files. (multiple values allowed)
  --main=MAIN                   Elements with this name prefix will be first in tree.
  --internal                    Include elements marked as @internal.
  --php                         Generate documentation for PHP internal classes.
  --skip-doc-path=SKIP-DOC-PATH Files matching this mask will be included in class tree, but will not create a link to their documentation. (multiple values allowed)
  --no-source-code              Do not generate highlighted source code for elements.
  --template-theme=TEMPLATE-THEME ApiGen template theme name. [default: "default"]
  --template-config=TEMPLATE-CONFIG Your own template config, has higher priority templateTheme.
  --title=TITLE                 Title of generated documentation.
  --todo                        Generate documentation for elements marked as @todo.
  --tree                        Generate tree view of classes, interfaces, traits and exceptions.
  -h, --help                    Display this help message.
  -q, --quiet                   Do not output any message.
  -V, --version                 Display this application version.

Help:
  Generate API documentation

```

## Tags

ApiGen так же как и PhpDoc используют теги для того что бы сослаться на информацию. Теги **@author**, **@version**, **@since**, **@copyright** и **@license** поддерживаются стандартным образом. Тега **@filesource** нет, так как программный код включается в документацию автоматически (точнее, в соответствии со значением конфигурационной опции документатора `--source-code`, а уже она по умолчанию yes).

Что бы показать результат генерации документации, файла с несколькими функциями будет мало, по этому для демонстрации была взята директория backend framework yii2.

Данные есть во вложении.

## Вывод по изученному материалу

### Doxygen

Первой мы рассмотрели программу Doxygen она популярна в использовании и проста в применении.

Не смотря на то, что данная программа не была создана для PHP она не уступает другим, которые были специализированы на данном языке.

Графический интерфейс делает простым управление и позволяет настроить результат так как нам необходимо, так же присутствует множество дополнительных функций и сама программа является универсальной так как может поддерживать не только язык PHP.

### PHPDoc

Второй инструментарий который был рассмотрен, это PHPDoc.

Я не могу сказать что данная программа является хорошей в использовании, она выглядит сырой и чем больше мы с ней работаем, тем больше это замечаем.

Программой тяжело пользоваться имея PHP версии 7.0+, это свидетельствует множество комментариев о попытке установить ее, так же, есть теги которые находятся в доработке и не работают во второй версии приложения.

В ней так же множество плюсов, она проста в использовании, так же как и doxygen может генерировать графическое описание.

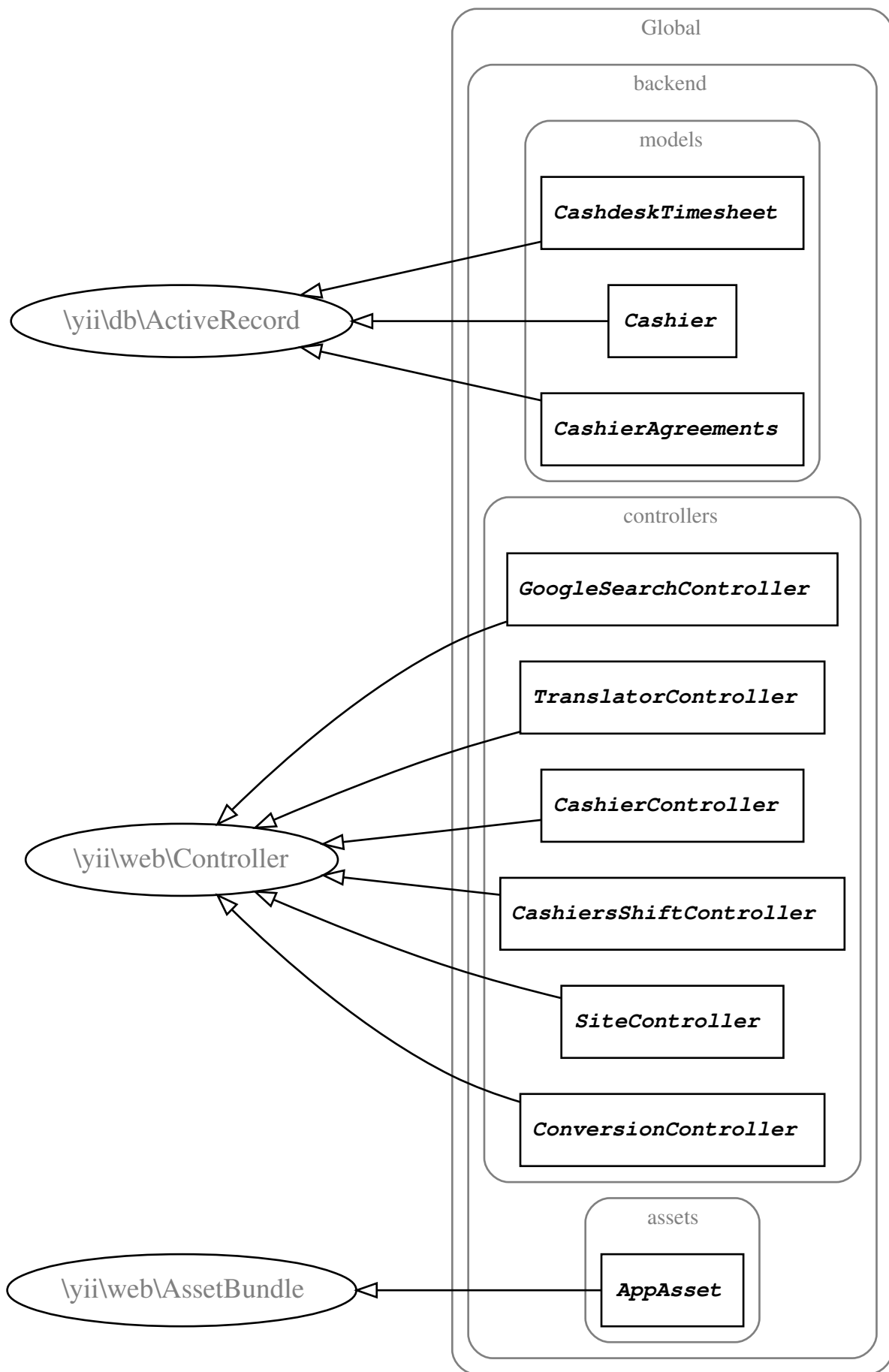
Вот пример backend/ , который был создан. (График находится на следующей странице).

### ApiGen

Простая программа для документации, очень похожа по работе с PHPDoc.

Проста в установке и легко устанавливается на NetBeans, в принципе, рекомендую работать с ней если используется данный инструментарий для кодирования. Так как, NetBeans позволяет многое авто документировать, к примеру те же самые теги, он предоставляет весь список их с описанием.

Для более детальной обработки документации, следует тщательно описать код. (Мне не хотелось копипастить чужой пример описание работы с данной программой по этому ссылку на пост я оставляю [тут](#)).



## Список информации взятой в сети

1. <https://phpdoc.org/> - сайт продукта
2. <http://www.apigen.org/> - сайт продукта
3. <http://www.stack.nl/~dimitri/doxygen/> - сайт продукта
4. [https://ru.wikipedia.org/wiki/%D0%93%D0%B5%D0%BD%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80\\_%D0%B4%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D0%B8](https://ru.wikipedia.org/wiki/%D0%93%D0%B5%D0%BD%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80_%D0%B4%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D0%B8) — статья о Генерации документации
5. <https://habrahabr.ru/post/134916/> - Hab, статья об использовании NetBeans и PHPdoc
6. <http://stackoverflow.com/> - помощник при установке и решении проблем :)
7. <https://habrahabr.ru/post/252101/> - Hab, статья о документации с помощью Doxygen
8. <https://github.com/phpDocumentor/phpDocumentor2> — Git установка PHPDoc
9. <https://github.com/ApiGen/ApiGen> — Git установка ApiGen
10. <http://appossum.com/appsite/techdetail/php-doc03> — статья по использованию и описанию ApiGen
11. <https://www.stack.nl/~dimitri/doxygen/manual/index.html> — Doxygen manual
12. <https://manual.phpdoc.org/HTMLframesConverter/default/> - PHPDoc manual.