Оглавление

Коротко о документе

Данный проект на стадии сверх активного развития и передокументирования, поэтому информация обновляется / дополняется каждый день.

Этот документ был сгенерирован 2017-10-04 в 16:39.

Данный проект изначально задумывался как панель управления Фитнесс клубом, абонементы, учет поминутных услуг и ведение бара. Но не успела завершиться подготовка первого альфа релиза, как проект перерос в нечто большее, и продолжает обрастать различными фишками.

Глава 1

Структура проекта

Проект делится на несколько основных частей.

1. DataBase
2. GUI Application
3. Web Application
4. Console Application's

Глава 2

Структура кодовой базы

2.1 API.Interfaces

2.1.1 Базовые классы

2.1.1.1 Interfaces.Config

В данном модуле располагаются классы по работе с конфигурационными файлами 4GE. Примеры и описание данных файлов Вы можете найти в разделе *Конфигурационные файлы*.

План:

1. Добавить возможность принудительного выбора изменяемого конфигурационного файла.
2. Добавить авто создание конфигурационного файла по шаблону.

2.1.1.1.1 Функции модуля

`class Config`

Класс отвечающий за работу с конфигурационными файлами.

Пример конфигурационного файла:

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
    <group_name group_attr_one="0" group_attr_two="GroupAttrValue">
        <section_name_one section_attr_one="SectionAttrValue"/>
        <section_name_two section_attr_two="SectionAttrValue" section_attr_one="3" />
    </group_name>
</configuration>
```

Пример использования:

```
config = Config()
config.set("database", "mysql", "server", "10.19.2.21")     // save value
config.set("database", "mysql", "username", "root")         // save value
```

```
config.get("database", "mysql", "server", "127.0.0.1")      // get value, return 10.19.2.
↪21
config.remove("database", "mysql", "username")
config.save()
```

fix_parms(*fn*)
Декоратор.

Используется для переопределения возвращаемого значения из **get** функции. И переопределяем следующие ключевые слова:

1. **$path** - Изменяет на путь до основного каталога проекта, где и лежит **config.xml**.
2. **$apache** - Изменяет на путь до основного каталога на веб сервере.

Результат Измененное значение функции.

Тип результата String

get(*group*, *item*, *attribute*, *value=''*)
Функция получения значения из конфигурационного файла. Если существует webroot/.4gain.etc/config.xml, то поиск изначально идет в нем, далее переходим на основной конфигурационный фаил.

Параметры

- `group` (*String*) – Имя группы в конфигурационном файле.
- `item` (*String*) – Имя секции в конфигурационном файле. Если имя секции опущено, идет поиск атрибута у группы.
- `attribute` (*String*) – Имя аттрибута секции в конфигурационном файле
- `value` (*String*) – Дефолтное значение, испоьуется если не найдено искомое, по умолчанию «».

Результат Искомое значение или дефолтное значение.

Тип результата String

set(*group*, *item*, *attribute*, *value=''*)
Функция записи значения в конфигурационного файла. Если существует webroot/.4gain.etc/config.xml, то запись идет в него, основной фаил игнорируется.

Параметры

- `group` (*String*) – Имя группы в конфигурационном файле.
- `item` (*String*) – Имя секции в конфигурационном файле. Если имя секции опущено, идет поиск атрибута у группы.
- `attribute` (*String*) – Имя аттрибута секции в конфигурационном файле
- `value` (*String*) – Значение параметра для записи, по умолчанию «».

Результат None, функция ничего не возвращает.

Тип результата Nothing

remove(*group*, *item*, *attribute*)
Функция удаления значения в конфигурационного файла. Если существует webroot/.4gain.etc/config.xml, то запись идет в него, основной фаил игнорируется. И удалить что либо из основного конфига не возможно.

Параметры

- `group` – Имя группы в конфигурационном файле.

- `item` – Имя секции в конфигурационном файле. Если имя секции опущено, идет поиск атрибута у группы.
- `attribute` – Имя аттрибута секции в конфигурационном файле

Результат None, функция ничего не возвращает.

Тип результата Nothing

`save()`
Функция записи конфигурационного файла на диск. Сохраняются оба конфигурационных файла.

Результат None, функция ничего не возвращает.

Тип результата Nothing

2.1.2 Подгруппы

2.1.2.1 Interfaces.MySQL

2.1.2.1.1 Базовые классы

2.1.2.1.1.1 Interfaces.MySQL.SQLAlchemy

Сервис модуль, который по сути не несет смысловой нагрузки, и предназначен в первую очередь для украшательства. В него вынесены все импорты из модуля SQLAlchemy, и далее мы импортируем только его, не внося гиганских портянок в модули схемы.

2.1.2.1.1.2 Функции модуля

`class Schema`

`static translate(value)`
Функция заглушка, выдает входящее значение без изменения, необходимо для SqlAlchemy.

Параметры `value` (*String*) – Значение

Результат ВХодящее значение без изменений

`__module__ = 'Interfaces.MySQL.SQLAlchemy'`

2.1.2.1.1.3 Interfaces.MySQL.Types

Модуль с дополнительными типами значений для схемы.

2.1.2.1.1.4 Функции модуля

`class PasswordType(*args, **kwargs)`

Базовые классы: `sqlalchemy.sql.type_api.TypeDecorator`

`impl = String(length=40)`

`bind_expression(bindvalue)`
Apply a SQL expression to an incoming cleartext value being rendered as a bound parameter.

For this example, this handler is intended only for the INSERT and UPDATE statements. Comparison operations within a SELECT are handled below by the Comparator.

class comparator_factory(*expr*)

Базовые классы: sqlalchemy.sql.sqltypes.Comparator

__eq__(*other*)
Compare the local password column to an incoming cleartext password.

This handler is invoked when a PasswordType column is used in conjunction with the == operator in a SQL expression, replacing the usage of the «bind_expression()» handler.

__module__ = 'Interfaces.MySQL.Types'

__module__ = 'Interfaces.MySQL.Types'

2.1.2.1.1.5 Interfaces.MySQL.Utils

Сервисный модуль, с набором утилит.

2.1.2.1.1.6 Функции модуля

transliterate(*string*)
Функция транслитиризации.

Параметры string (*String*) – Строковое значение на русском языке

Результат Строковое значение в ANSII

Тип результата String

safe_string(*unsafe_string*)
Функция фильтр, очищает строковое значение от спец символов. Безопасные символы это буквы, цифры и «-«, «_».

Параметры unsafe_string (*String*) – Не безопасное строковое значение

Результат Безопасное строковое значение

Тип результата String

2.1.2.1.2 Подгруппы

2.1.2.1.3 Функции модуля

init(*config*)

init_fast(*config*)

2.1.3 Функции модуля

Глава 3

Дополнения и разный хлам

Глава 4

Документация и утилиты

4.1 Консольные приложения

Проект содержит множество консольных утилит как для непосредственной работы, так и для облегчения разработки.

4.1.1 База данных

4.1.1.1 4gain-db-backup.sh

Место запуска: `Any`. Скрипт: `4gain-db-backup.sh`.

```
# Запуск
./4gain-db-backup.sh;
```

4.1.1.2 4gain-db-restore.sh

Место запуска: `Any`. Скрипт: `4gain-db-restore.sh`.

```
# Запуск
./4gain-db-restore.sh;
```

4.1.2 Установка

4.1.2.1 4gain-install-devel.sh

Место запуска: `Devel Station`. Скрипт: `4gain-install-devel.sh`.

```
# Запуск
./4gain-install-devel.sh;
```

4.1.2.2 4gain-install-gui.sh

Место запуска: `Manager Station`. Скрипт: `4gain-install-gui.sh`.

```
# Запуск
./4gain-install-gui.sh;
```

4.1.2.3 4gain-install-web.sh

Место запуска: `Web Application Server`. Скрипт: `4gain-install-web.sh`.

```
# Запуск
./4gain-install-web.sh;
```

4.1.2.4 4gain-requirements-update.sh

Место запуска: `Devel Station`. Скрипт: `4gain-requirements-update.sh`.

Скрипт обновления `requirements.txt` исходя из установленных компонентов окружения.

```
# Запуск
./4gain-requirements-update.sh;
```

4.1.3 Запуск

4.1.3.1 4gain-run-gui.sh

Место запуска: `Devel Station` | `Manager Station`. Скрипт: `4gain-run-gui.sh`.

```
# Запуск
./4gain-run-gui.sh;
```

4.1.3.2 4gain-run-web.sh

Место запуска: `Devel Station`. Скрипт: `4gain-run-web.sh`.

```
# Запуск
./4gain-stable-web.sh;
```

4.1.4 Web Сервер

Утилиты создания окружения для Apache, сборки документации Sphinx.

4.1.4.1 4gain-server.py

Место запуска: `Web Application Server`. Скрипт: `4gain-server.py`.

```
# Запуск
./4gain-server-web.py;
```

4.1.4.2 4gain-sphinx-update.sh

Место запуска: `Devel Station`. Скрипт: `4gain-sphinx-update.sh`.

Скрипт сборщик автодокументации Sphinx и построения графиков зависимостей SQL.

```
# Запуск
./4gain-sphinx-update.sh;
```

4.1.4.3 4gain-md-update.py

Место запуска: `Devel Station`. Скрипт: `4gain-md-update.py`.

```
# Запуск
./4gain-md-update.py;
```

4.1.5 Утилиты развертывания

Утилиты отделения «мух от котлет». На рабочей системе нет необходимости хранить весь набор данных проекта. Данные утилиты позволяют подготовить обособленное окружение очищенное от лишних файлов.

Также данные скрипты применяются для разделения ветки разработчика от продакшена.

4.1.5.1 4gain-stable-gui.sh

Место запуска: `Web Application Server`. Скрипт: `4gain-stable-gui.sh`.

Утилита разделения кодовой базы, выделяет из общего дерева проекта `/home/engine` файлы необходимы для работы GUI приложения и переносит их в `/home/engine.gui` для дальнейшего деплоя на рабочие станции.

```
# Запуск
./4gain-stable-gui.sh;
```

4.1.5.2 4gain-stable-web.sh

Место запуска: `Web Application Server`. Скрипт: `4gain-stable-web.sh`.

Утилита разделения кодовой базы, выделяет из общего дерева проекта `/home/engine` файлы необходимы для работы Web приложений и переносит их в `/home/engine.web`.

Перенос происходит по следующей схеме:

1. Остановка Web сервера.
2. Размонтирование всех виртуальных каталогов Web приложений.
3. Перенос файлов.
4. Монтирование виртуальных каталогов Web приложений.
5. Запуск Web сервера.

```
# Запуск
./4gain-stable-web.sh;
```

4.1.5.3 4gain-stable-web-vhost.sh

Место запуска: `Devel Station`. Скрипт: `4gain-stable-web-vhost.sh`.

Утилита переноса данных разработки web приложения в продакшен.

```
# Example:
#   ./4gain-stable-web-host.sh pro-4gain-ambclub ru-ambclub

# Запуск
./4gain-stable-web-host.sh <vhost_src_name> <vhost_dst_name>;
```

4.2 Конфигурационные файлы

Конфигурационные файлы хранятся вот в такой файловой структуре:

```
/home/engine/config.xml # (4GE Config)
/home/apache/pro-4gain/.4gain.etc/config.xml # (4GW Config для 4gain.pro)
/home/apache/pro-4gain-engine/.4gain.etc/config.xml # (4GW Config для engine.4gain.pro)
```

4.2.1 Конфигурационный фаил 4GE

Пример «классического» конфигурационного файла:

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
    <database use_synchronizer="0" use_inicialise="0">
        <mysql database="db_4gain" password="<password>" server="10.19.2.20" user="<user>" charset="utf8" />
    </database>
    <logging use_file="1" use_syslog="0" >
        <console level="10" />
        <splash level="20" />
        <file level="10" path="$path/var/log/4gain.log" when="d" interval="1" count="7"/>
        <syslog level="30" address_ip="10.19.2.18" address_port="514" />
    </logging>
    <session use_autologin="1">
        <installation uuid="64d5c56c-c938-4198-9e80-ce8fea3810ed" />
        <autologin username="v.tatarnikov" password="test" club_select="2" />
    </session>
    <gui gtk-major="3" gtk-minor="16" gtk-micro="7" date-format="%Y.%m.%d" use_splash="1">
        <Main glade="$path/share/gui/Main.glade" image-auth-logo="$path/share/icons/target.png"
              image-logo="$path/share/icons/worldwide.png" />
        <MainSplash glade="$path/share/gui/MainSplash.glade" image-splash="$path/share/images/splash.png" />

        <UserManager glade="$path/share/gui/UserManager.glade" image-logo="$path/share/icons/user-17.png" />
        <UserProfile glade="$path/share/gui/UserProfile.glade" image-logo="$path/share/icons/user-4.png" />
        <UserProfilePassword glade="$path/share/gui/UserProfilePassword.glade" image-logo="$path/share/icons/user-5.png" />

        <SaleManager glade="$path/share/gui/SaleManager.glade" />

        <ProductManager glade="$path/share/gui/ProductManager.glade" />
        <ProductManagerDataEntrace glade="$path/share/gui/ProductManagerDataEntrace.glade" />
        <ProductManagerDataPrice glade="$path/share/gui/ProductManagerDataPrice.glade" />
        <ProductManagerDataGroup glade="$path/share/gui/ProductManagerDataGroup.glade" />
```

```
        <ProductManagerDataCombinate glade="$path/share/gui/ProductManagerDataCombinate.glade" />

        <ProductManagerEntrace glade="$path/share/gui/ProductManagerEntrace.glade" />
        <ProductSale glade="$path/share/gui/ProductSale.glade" />

        <CashdeskManager glade="$path/share/gui/CashdeskManager.glade" />

        <OperdayManager glade="$path/share/gui/OperdayManager.glade" image-logo="$path/share/icons/dollar-symbol-1.png" />
        <OperdayManagerOpen glade="$path/share/gui/OperdayManagerOpen.glade" image-logo="$path/share/icons/piggy-bank.png" />
        <OperdayManagerClose glade="$path/share/gui/OperdayManagerClose.glade" image-logo="$path/share/icons/piggy-bank-1.png" />
        <OperdayReport glade="$path/share/gui/OperdayReport.glade" image-logo="$path/share/icons/pie-chart.png" />

        <ClubCurrentInformation glade="$path/share/gui/widgets/ClubCurrentInformation.glade" />
        <CashdeskCurrentInformation glade="$path/share/gui/widgets/CashdeskCurrentInformation.glade" />
        <UserCurrentInformation glade="$path/share/gui/widgets/UserCurrentInformation.glade" />
    </gui>
</configuration>
```

4.2.2 Конфигурационный фаил 4GW

Данный конфигурационный фаил используется только в среде 4GW, и нигде более. Он переопределяет параметры основного конфигурационного файла 4GE, для каждного виртаульного хоста (Apache Virtual Host).

Пример «классического» конфигурационного файла:

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
    <web site_id="1">
        <application import_name="forGain"
                     static_url_path="/static"
                     static_folder="/home/apache/pro-4gain-www/.4gain.web/static"
                     static_folder_master="/home/apache/pro-4gain-www/.4gain/share/static"
                     template_folder="/home/apache/pro-4gain-www/.4gain.web/templates/default"
                     template_folder_master="/home/apache/pro-4gain-www/.4gain/share/templates/default"/>

        <!-- Использовать ли внишние аналитические сервисы (применяется в шаблонах) -->
        <analytics-google enable="True" />
        <analytics-yandex enable="True" />

        <!-- Идентификатор клуба при использовании на сайте информации о "клубе" Interfaces.MySQL.Schema.Club [1] и WEB.Module.Club [2] -->
        <module-club club_id="1" />

        <!-- Определение ключа к функциям google api, применяемых к примеру в WEB.ModuleContact [3] -->
        <google-api gmap="AIzaSyBUxH5_DlaI8DpyfOR-wVPlb_nOFKNR-ew" />
    </web>
</configuration>
```

Дополнительную информацию по использованию второго конфигурационного файла можно посмотреть в исходном коде *Interfaces.Config*, а также в module-web-application.

4.3 Список предстоящих задач

План:

1. Добавить возможность принудительного выбора изменяемого конфигурационного файла.
2. Добавить авто создание конфигурационного файла по шаблону.

исходный элемент

Глава 5

Примеры использования кодовой базы

1. 4gain.pro - Основная сайт визитка
2. engine.4gain.pro - Описание возможностей кодовой базы, примеры
3. ambclub.ru - Сайт ЦСД Амбассадор. В нем используется наибольшее количество функций 4GE (4Gain Engine). Помимо 4GW (4Gain Web), в клубе используется 4GG (4Gain GUI) под Fedora Linux.
4. betar-siberia.ru - Сайт ПФК Бетар-Сибирь. Легкий портал, со статьями, каталогом и контактами.
5. uchetenergo.com - Сайт сети магазинов «Счетсчики».

Глава 6

Поиск и индексы

- genindex
- modindex
- search

Содержание модулей Python

i

Алфавитный указатель

Symbols

B

C

F

G

I

P

R

S

T