xadrpy Documentation

Release 0.6.dev20120820131342

Palánkai Csaba

August 20, 2012

Tartalomjegyzék

This documentation describes *xadrpy* version 0.6.

1. fejezet

Beginning

1.1 Telepítés

```
pip install xadrpy
```

1.1.1 Függőségek

- Python 2.5 (vagy újabb, de 2.x).
- Django 1.4.x
- *argparse* 1.2.1 (vagy újabb)
- django-mptt 0.5.2

1.1.2 Adatbázis

Leginkább a PostgreSQL vagy a MySQL használatát javasolom.

Az optimális és biztonságos működés érdekében, javaslom:

- Külön felhasználó létrehozását az egyes *xadrpy* telepítésekhez.
- Külön adatbázis létrehozását az egyes *xadrpy* telepítésekhez.

1.2 Kódolási irányelvek

Ennek az oldalnak a célja, hogy bemutassa az általam lefektetett kódolási irányelveket, amelyeket a kód teljes egészében alkalmazok.

1.2.1 Verziók

> **Warning:** A kód az 1.Y verzió eléréséig beta állapotban van, ami annyit takar, hogy az egyes kiadásokban is lehetnek nagyobb különbségek.

Az xadrpy kiadásai során az alábbi konvenciókat követem.

X Fő verzió (például (0.Y, 1.Y) - a különböző fő verziók egymáshoz képest átfogó különbségeket is tartalmazhatnak.

X.Y Verzió (például 0.4, 1.0) - Az egyes verziók egymáshoz képest nagyobb mérvű átalakításokat is tartalmazhatnak, de szerkezetileg egymásra épül.

X.Y.Z Release - kiadás.

Egyesmáshoz képes nem tartalmaz függőségeket, nem módosul az adatbázis felépítés.

1.2.2 Appok elhelyezkedése

Note: *django* terminológia szerint az egyes önálló egységek neve: application (app).

Az alap csomag az *xadrpy*, ez alatt találhatóak az egyes fő app-ok (core). Ilyen core app például az *api*, *workers*, stb.

Note: Néhány kiegészítő funkció miatt az *xadrpy* funkcionál mint app.

A kiegészítő appok például a *contrib* illetve a *social* alatt találhatóak.

A *vendor* alatt azon az *xadrpy* részeként használható appok, csomagok találhatóak, amelyek legfeljebb minimális módosítással kerültek be a keretrendszerbe, vagy csak arra építkezik.

1.2.3 Egyes appok felépítése

Modulok

- *__init__.py* - publikus interface
- *defaults.py* - Alapértékek
- *conf.py* - Csomag beállításai, alapértékei
- *base.py* - Alap funkciók, a csomag tetszőleges része importálhatja
- *managers.py - adatelérési réteg (DAL)*
- *models.py* - adatbázis modellek
- *libs.py - Üzleti logika (BLL)*
- *urls.py* - URL felépítés
- *admin.py - django admin* felülethez interface
- *views.py* - view függvények, osztályok
- *xtensions.py - xadrpy* kiterjesztések
- *middleware.py* - Request middleware osztályok
- *signals.py* - Publikus szignálok
- *context_processors.py* - Template context kiterjesztések

Csomagok

- *templatetags* - az app által publikált template tag-ek
- *migrations* - Adatbázis migrációk
- *management.commands* - A csomag által publikált django-admin parnacsok

Almappák

- *templates* - A csomag template fájljai
- *static* - A csomag

Note: Az egyes modulok opcionálisak, nem feltétlen implementálja minden csomag.

Note: A felépítés tervezésekor figyelembe vettem a *django* irányelveit.

1.3 xtensions.py approach

The best place for use xadrpy extras, define your own plugins, tools is the xtension.py of your module. This contains xadrpy or xadrpy module configuration. The xadrpy autodiscover (when application started - before first request) traverses all xtensions.py if the package of xtensions.py in listed on INSTALLED_APPS.

1.3.1 Project level xtensions.py

You can define project level xtensions.py in your settings.

```
XTENSIONS = "<project_name>.xtensions"
```

1.3.2 Loading order

The applications order in INSTALLED_APPS gives the primary loading order. The autodiscover load project xtensions.py after load applications xtensions.py, so you can redefine, reset some values in that.

2. fejezet

Core packages

2.1 Theming

Theming package provides a solution for theme handling.

2.1.1 Approach

- You can define one or more themes (with layouts and skins).
- The Theming recognizes these definitions.
- Provides selecting layouts, skins

Background

You can reference the layout with ‘@layout_name‘formula. The *Loader* resolve ‘@layout_name‘and ensure proper layout html file. The context processor set *theming_layout* variable to base / selected layout.

2.1.2 Installation

```
TEMPLATE_LOADERS = (
        'xadrpy.core.theming.loaders.Loader', #Be the first
        ...
)

TEMPLATE_CONTEXT_PROCESSORS = (
        ...
        'xadrpy.core.theming.context_processors.theming',
        ...
)

INSTALLED_APPS = (
        ...
        'xadrpy',
        ...
        'xadrpy.core.theming',
        ...
)
```

2.1.3 Theme definition

```
THEME = {
        # description elements
        'title': "Title of the theme",
        'description': 'Description of the theme. (Long plain text)',
        'thumbnail': 'relative path in static directory of image - it will be resized',
        'translation': {
                'en': {                                                         #language code
                        'title': "Title of the theme for this language",
                        'description': 'Description of the theme. (Long plain text) for this language
                        'thumbnail': 'relative path in static directory of image - it will be resized
                },
                'hu': {                                                         #language code
                                                                                        #Without tit
                        'description': '',                              #Disable description in this
                },
        },

        # theme elements
        'doctype': ['xhtml','xhtml-strict'],    #list of doctypes
        'static_path': "",                                              #relative path in STATIC dir
        'template_path': "",                                    #relative path in TEMPLATES direcori
        'layouts': (                                                    # you can leave blank (or do
                ("vehicle", {                                           #The reference name of the l
                        "title": "",                                    #you can give description el
                        "file: "",                                              #template file relat
                        "rewrite": {
                                                                                        #you can giv
                                        },
                        }),
                ),
        'skins':(                                                               #you can leave empty
                ("base", {                                                      #The reference name
                        "title: "",                                             #you can give descri
                        "styles": (
                                {"file": "style.css", "media": "all"},
                                {"file": "style2.css", "condition": "if IE"},
                        )
                        "scripts": (
                                "skin-scripts.js",                      #relative path for required
                        ),
                        "rewrite": {
                                                                                        #you can giv
                        },
                }),
        ),
        'base': "base.html",                                    #base layout
        "rewrite":{
                "page.html": "my-page.html",            #you can give some redefinitions for templat
                }
        }
```

Rewrite order:

- main rewrite

- layout rewrite
- skin rewrite

2.1.4 Registering

```
# in your xtensions.py
from xadrpy.theming.libs import theme_store

THEME = {... see above ...}

theme_store.register(THEME)
```

2.1.5 Using

The context processor set the selected (with route/page...) layout to *theming_layout* context var. You can redefine it from your views.

```
{% extends "theming_layout" %}
...

{% extends "@layout_name" %}
...
```

or

```
render_to_response("@layout_name", ...)
...
```

but the best solution: define a base.html in your template root and it contains just one line: {% extends „theming_layout” %}. You can use proper „base.html”.

3. fejezet

Contrib packages

4. fejezet

External (Vendor) packages

5. fejezet

Index

- *genindex*
- *modindex*
- *search*