

DE LA RECHERCHE À L'INDUSTRIE



Django et PyQt4 : technologies Python pour la simulation en astrophysique



CHAPON Damien - damien.chapon@cea.fr

www.cea.fr

<http://irfu.cea.fr/Projets/COAST>

Journées Informatique de l'IRFU-IN2P3.

JI 2016 – 26-29 Septembre 2016 – Le Lioran, Cantal, France.



Django : application web CoastDB

Motivation et existant

Architecture

Création d'une application web Django

Modèle de données et ORM

Configuration d'url, vues et gabarits

PyQt4 : IHM PyQt4 de post-traitement (Horus)

Présentation

Design d'IHM et Qresources (QtDesigner → Python)

Personnalisation de l'application : QStyle

Import de données dans CoastDB

1. CoastDB : application web Django de publication de simulations numériques en astrophysique



Mutualisation des données de simulation en astrophysique

- Volume de données : ~100 To / simulation.
- Coût du temps de calcul alloué (~10 Mh / projet).
- Nécessité d'imposer la réutilisation des données de simulation par plusieurs équipes scientifiques. Exigence requise par :
 - ▶ Organismes de financement (ANR, ERC, ...),
 - ▶ Commissions d'allocation de temps de calcul (centre de calculs régionaux, nationaux ou européens).
- Objectif : maximiser les retombées scientifiques des études numériques (cst. ressources HPC).





Mutualisation des données de simulation en astrophysique

- Volume de données : ~100 To / simulation.
- Coût du temps de calcul alloué (~10 Mh / projet).
- Nécessité d'imposer la réutilisation des données de simulation par plusieurs équipes scientifiques. Exigence requise par :
 - ▶ Organismes de financement (ANR, ERC, ...),
 - ▶ Commissions d'allocation de temps de calcul (centre de calculs régionaux, nationaux ou européens).
- Objectif : maximiser les retombées scientifiques des études numériques (cst. ressources HPC).

Principaux freins à la mutualisation

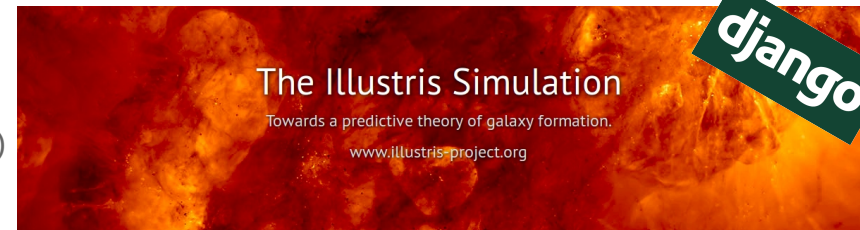
- Spécificité des codes de calculs (solveurs, paramètres de config., ...),
- Spécificité des modèles numériques,
- Format de données non standardisés.





Le projet Illustris (www.illustris-project.org)

- API web d'accès aux données (scripts Python/Matlab/IDL)
- Contenu téléchargeable + accès aux données brutes,
- Formulaire de recherche dans un catalogue de halo de matière noire



Public Data Access [Overview](#) / [Subhalo Search](#)

You can search on min/max ranges for one or more Subhalo fields. When you select a field, the minimum, mean, and maximum at this snapshot are calculated for reference. The icons on each row link to: full details in the browsable API (spyglass), merger tree visualization (tree), galaxy stellar images (picture), snapshot extraction (download arrow).

1. Select simulation: and snapshot: or enter snapshot number:

2. Select search fields and enter bounds:

Field	>=	<=	min	mean	max
<input type="text" value="mass"/>	<input type="text" value="10.0"/>	<input type="text" value="20.0"/>	0.131521	10.5845	25250.1

+ Add Search Field

Search

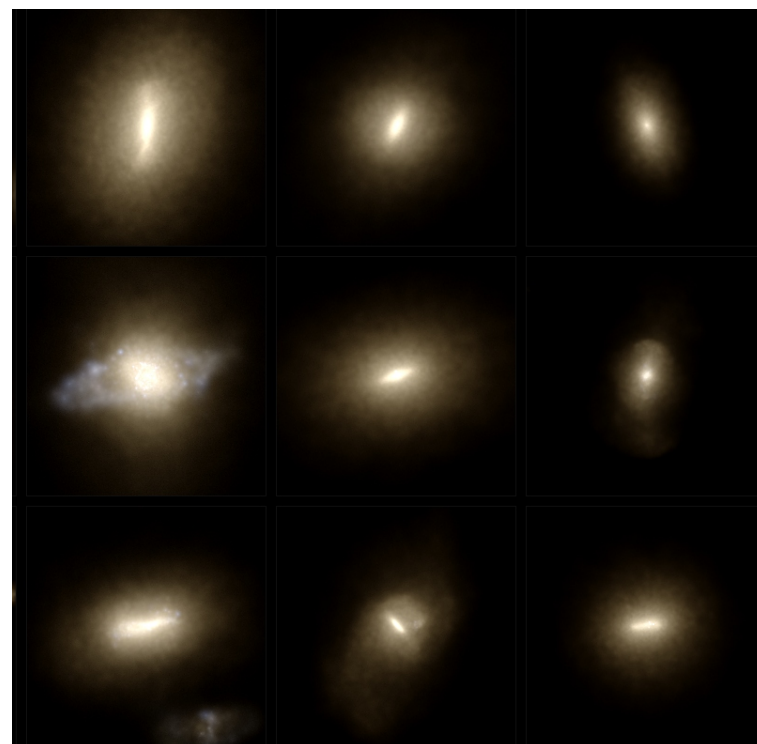
ID	M_{tot}	M_{tot}	M_*	M_{gas}	M_{DM}	x	y	z	$r_{1/2}$	SFR	$Z_*(2r_{1/2,*})$	$Z_{\text{gas}}(2r_{1/2,*})$	v_{max}	σ_v	m_U	m_B	m_R
	[$10^{10} M_{\odot}/h$]	[$\log M_{\odot}$]	[$10^{10} M_{\odot}/h$]	[$10^{10} M_{\odot}/h$]	[$10^{10} M_{\odot}/h$]	[ckpc/h]	[ckpc/h]	[ckpc/h]	[ckpc/h]	[M_{\odot}/yr]	-	-	[km/s]	[km/s]	[mag]	[mag]	[mag]
0	25300.0	14.55	162.0	3070.0	22000.0	900.4	26286.7	18321.9	675.6	4.1110	0.0224	0.00635	896.9	543.8	-23.67	-24.06	-25.26
1	2470.0	13.54	33.0	40.3	2390.0	181.1	24638.8	16896.9	336.5	1.8752	0.0273	0.0601	399.3	262.1	-22.04	-22.45	-23.64
2	406.0	12.76	18.8	0.771	386.0	826.9	26747.3	17363.4	57.2	0.3440	0.0194	0.027	393.6	243.4	-21.08	-21.53	-22.77





Le projet Illustris (www.illustris-project.org)

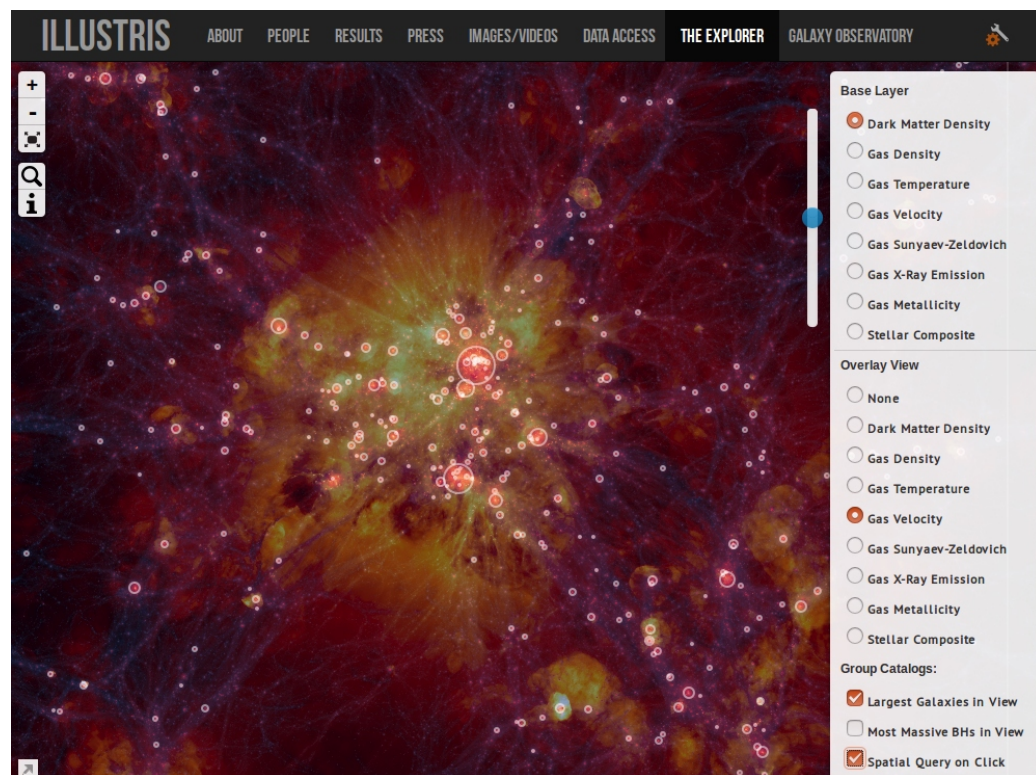
- API web d'accès aux données (scripts Python/Matlab/IDL)
- Contenu téléchargeable + accès aux données brutes,
- Formulaire de recherche dans un catalogue de halo de matière noire,
- Catalogues de galaxies (observables synthétiques)
 - ▶ Images stellaires de synthèse.





Le projet Illustris (www.illustris-project.org)

- API web d'accès aux données (scripts Python/Matlab/IDL)
- Contenu téléchargeable + accès aux données brutes,
- Formulaire de recherche dans un catalogue de halo de matière noire,
- Catalogues de galaxies (observables synthétiques)
 - ▶ Images stellaires de synthèse.
- Outils d'exploration et de visualisation :
 - ▶ Arbres de fusion des halos,
 - ▶ Formulaire d'extraction des particules.

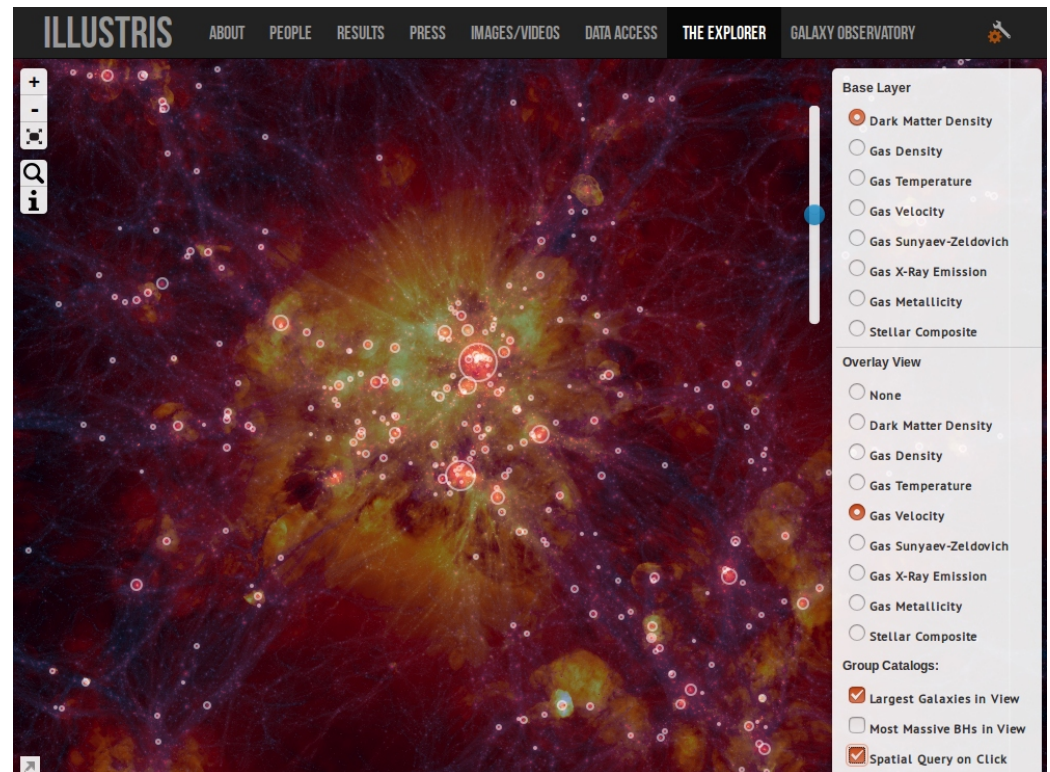
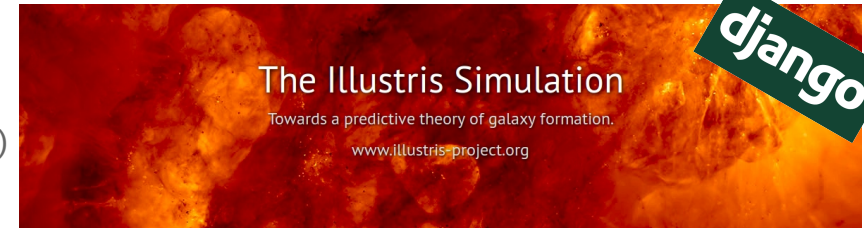




Le projet Illustris (www.illustris-project.org)

- API web d'accès aux données (scripts Python/Matlab/IDL)
- Contenu téléchargeable + accès aux données brutes,
- Formulaire de recherche dans un catalogue de halo de matière noire,
- Catalogues de galaxies (observables synthétiques)
 - ▶ Images stellaires de synthèse.
- Outils d'exploration et de visualisation :
 - ▶ Arbres de fusion des halos,
 - ▶ Formulaire d'extraction des particules.

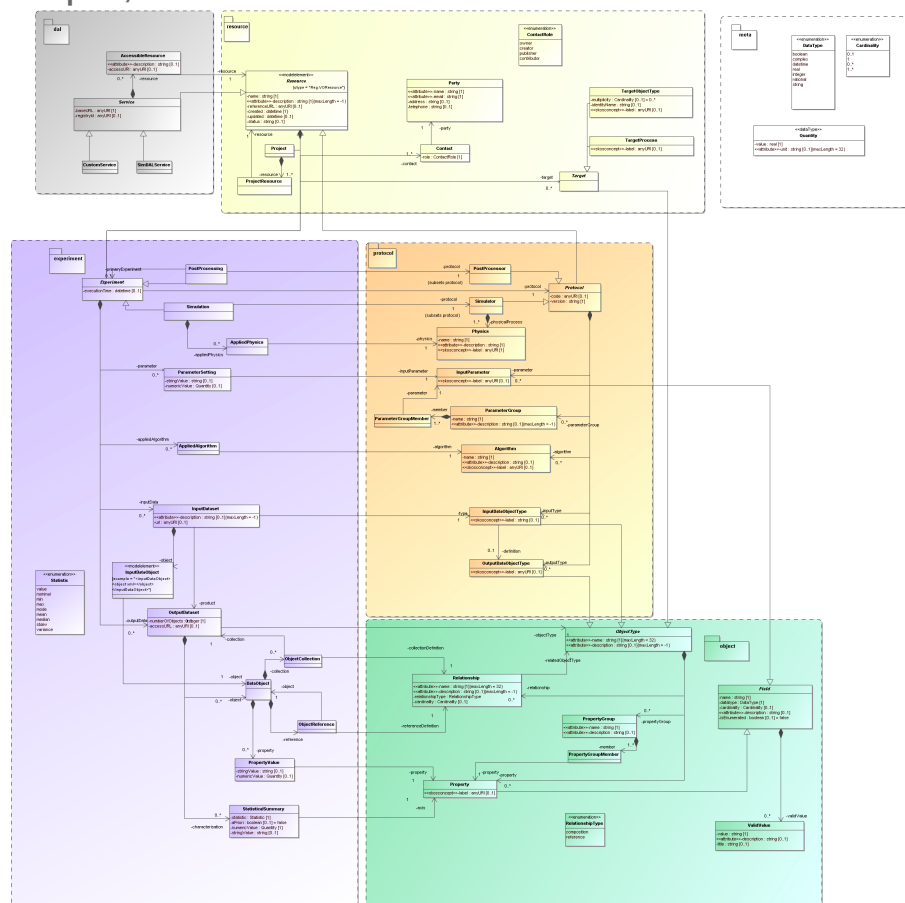
- ▶ **Projet unique – une seule simulation,**
- ▶ **Contenu statique,**
- ▶ **Approche orientée cosmologie.**

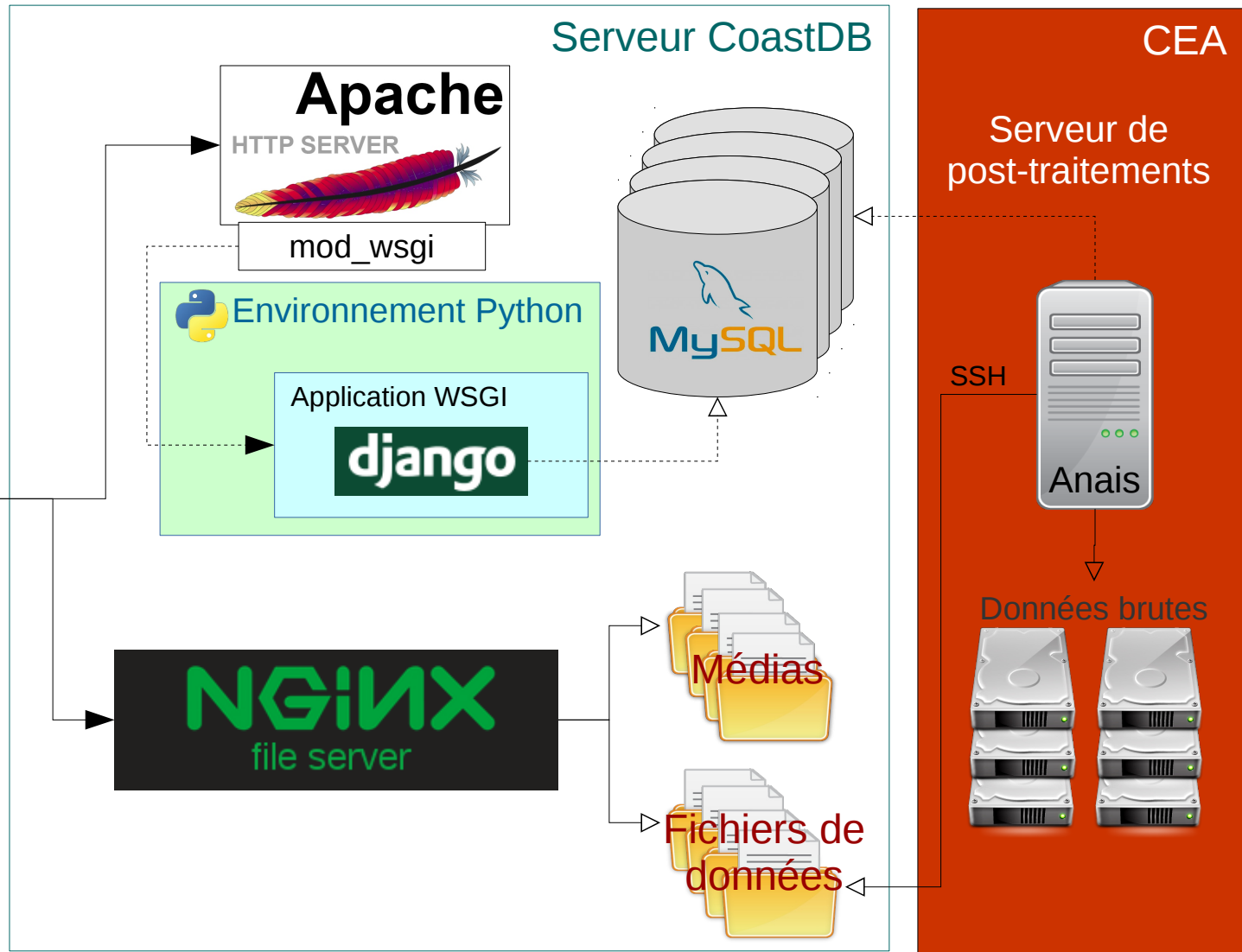




Base de données multi-simulations & multi-domaines

- Projets numériques divers (MHD solaire / Interactions étoile-planètes / Milieu inter-stellaire / Explosions de supernovae / Formation stellaire / Formation des galaxies / Cosmologie),
- Outils collaboratif de pré-publication scientifique,
- Outil de partage de données (OpenData),
- Inspiré du standard IVOA SimDM :
 - ▶ <http://www.ivoa.net/documents/SimDM>.
- Socle techno. : Django (Python)
 - ▶ <https://www.djangoproject.com>.







Django-admin : initialisation d'un projet/application

```
> django-admin startproject coastdb
> cd coastdb && ls
coastdb  manage.py
> python manage.py startapp astrowebapp
> ls astrowebapp/
admin.py  apps.py  __init__.py  migrations  models.py  tests.py  views.py
> ls coastdb/
__init__.py  settings.py  urls.py  wsgi.py
```





Django-admin : initialisation d'un projet/application

```
> django-admin startproject coastdb
> cd coastdb && ls
coastdb  manage.py
> python manage.py startapp astrowebapp
> ls astrowebapp/
admin.py  apps.py  __init__.py  migrations  models.py  tests.py  views.py
> ls coastdb/
__init__.py  settings.py  urls.py  wsgi.py
```

Configuration du projet

■ Module *settings* : [coastdb/settings.py](#)

```
# Application definition
INSTALLED_APPS = [
    'suit',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'astrowebapp.apps.AstrowebappConfig',
]
```





Django-admin : initialisation d'un projet/application

```
> django-admin startproject coastdb
> cd coastdb && ls
coastdb  manage.py
> python manage.py startapp astrowebapp
> ls astrowebapp/
admin.py  apps.py  __init__.py  migrations  models.py  tests.py  views.py
> ls coastdb/
__init__.py  settings.py  urls.py  wsgi.py
```

Configuration du projet

■ Module *settings* : [coastdb/settings.py](#)

```
# Database
# https://docs.djangoproject.com/en/1.10/ref/settings/#databases
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'COASTDB',
        'USER': 'damien',
        'PASSWORD': 'mys3cr3t#Pwd',
        'HOST': '127.0.0.1',
        'PORT': '3306'
    }
}
```





Définition des classes du modèle de données

■ Module *models* : [astrowebapp/models.py](#)

```
from django.conf import settings
from django.db import models
from django.utils.translation import ugettext as _
from ..utils.validators import alias_validator

class Category(models.Model):
    alias = models.CharField(max_length=16, verbose_name=_("alias"), primary_key=True,
                            validators=[alias_validator])
    cat_name = models.CharField(max_length=64, verbose_name=_("category label"), unique=True)

class Project(models.Model):
    creator = models.ForeignKey(settings.AUTH_USER_MODEL)
    name = models.CharField(max_length=128, verbose_name=_("name"))
    creation_date = models.DateTimeField(verbose_name=_("creation date"), auto_now_add=True)
    cat = models.ForeignKey(Category, verbose_name=_("project category"))
    alias = models.CharField(max_length=16, verbose_name=_("alias"), validators=[alias_validator])
    is_pub = models.BooleanField(default=False, verbose_name=_("published project ?"))

    class Meta:
        unique_together = ['cat', 'alias']
```

Enregistrement des classes du modèle de données

■ Module *admin* : [astrowebapp/admin.py](#)

```
from django.contrib import admin
from .models import Category, Project

admin.site.register(Category)
admin.site.register(Project)
```





Migrations du modèle de données

■ Création/mise à jour de la structure de la base de données

```
> python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, astrowebapp, sessions
Running migrations:
  Rendering model states... DONE
  Applying astrowebapp.0001_initial... OK
> python manage.py createsuperuser
# [...]
```

Lancement du serveur de développement

```
> python manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).
September 23, 2016 - 09:05:10
Django version 1.10.1, using settings 'coastdb.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

Django administration

Username:

Password:

▶ <http://127.0.0.1:8000/admin/>





Interface d'administration (django-suit)

■ Système de gestion de contenu (CMS) ergonomique pour créer/éditer :

- ▶ Des [Projets](#) numériques,
- ▶ Des runs de [Simulation](#), associés à des [Codes de calcul](#),
- ▶ Des [Résultats](#) et leurs [Fichiers de données](#) associés,
- ▶ Des [Données statistiques](#) et des [Catalogues](#) (données tabulées).

The screenshot shows the 'CoastDB administration' interface. The top navigation bar includes the site name, the current date and time (Wednesday, 21. September 2016, 16:54), and user information (Welcome, Damien. Change password | Log out). The main content area is titled 'Home » Coastdb » Project categories' and features a table of project categories. A green 'Add project category' button is visible in the top right of the table area. The table has columns for selection, category label, alias, project count, administrator, since version, and order. Below the table, there is a search bar, a 'Go' button, and a 'Save' button. The bottom of the table area shows '1 - 5 / 5 project categories'.

<input type="checkbox"/>	Category label	Alias	Projects	Administrator	Since version	Order	▼ ×
<input type="checkbox"/>	Solar Magnetohydrodynamics	MHD_SOL	1 🔍	Allan Sacha BRUN	1.0.0	↑ ↓	
<input type="checkbox"/>	Star-planet interactions	INT_STAR_PL	1 🔍	Allan Sacha BRUN	1.0.0	↑ ↓	
<input type="checkbox"/>	Star formation	STAR_FORM	1 🔍	Patrick HENNEBELLE	1.0.0	↑ ↓	
<input type="checkbox"/>	Galaxy formation	GAL_FORM	0	Frédéric BOURNAUD	1.0.0	↑ ↓	
<input type="checkbox"/>	Galactic mergers	GAL_MERGERS	1 🔍	Frédéric BOURNAUD	1.0.0	↑ ↓	





Interface d'administration (django-suit)

■ Système de gestion de contenu (CMS) ergonomique pour créer/éditer :

- ▶ Des [Projets](#) numériques,
- ▶ Des runs de [Simulation](#), associés à des [Codes de calcul](#),
- ▶ Des [Résultats](#) et leurs [Fichiers de données](#) associés,
- ▶ Des [Données statistiques](#) et des [Catalogues](#) (données tabulées).

CoastDB administration Friday, 3. June 2016 09:30 Welcome, Patrick. Change password | Log out

Home » Coastdb » Projects

Keyword Project category Published project ? Search Add project

2015 2016

<input type="checkbox"/>	Name	Project category	Simulations	Creation date	Published project ?	Order
<input type="checkbox"/>	Colliding flow simulations	Star formation	4	Dec. 4, 2015, 4:27 p.m.	✓	↑ ↓
<input type="checkbox"/>	Star-planet interaction dummy project	Star-planet interactions	0	June 1, 2016, 10:48 a.m.	✓	↑ ↓

0 of 2 selected Save

1 - 2 / 2 projects





Interface d'administration (django-suit)

Édition d'une catégorie

CoastDB administration
Welcome, **Damien**. [Change password](#) | [Log out](#)

[Home](#) » [Coastdb](#) » [Project categories](#) » **Star formation**

- [Home](#)
- [Authorization](#)
- [History](#)
- [Commons](#)
- [Commons versions](#)
- Project categories**
- [Algorithms](#)
- [Physical processes](#)
- [Input parameters](#)
- [Object types](#)
- [Physical units](#)

- [Projects](#)
- [Protocols](#)
- [Experiments](#)
- [Results](#)
- [Products](#)
- [Statistical data](#)

Category information

Alias:	STAR_FORM <small>Define a unique project category key here.</small>
Category label: *	<input style="width: 90%;" type="text" value="Star formation"/> <small>Enter the displayed category name here.</small>
Reference URL:	<input style="width: 90%;" type="text" value="http://"/>
Administrator: *	<input style="width: 90%;" type="text" value="Patrick HENNEBELLE"/>
Since version:	1.0.0

Summary

The CoastDB database contains results of heavy numerical simulations computed in order to study the problem of star formation, essentially molecular cloud formation, evolution and collapse. Understanding the dynamical evolution of the interstellar medium (ISM) and its relation to stellar birth is a key challenge in astronomy and astrophysics. The CoastDB project aims at providing observers and theorists studying formation and evolution of molecular clouds, their morphological and kinematical characteristics, and the formation of stars in their interior with a set of theoretical tools and a database of models to aid in the analysis and interpretation of current and future observations.

[Delete](#)

Tools

- [History](#)
- [View on site](#)
- [Add project category](#)





Définition de l'url associé à une instance

■ Module *models* : [astrowebapp/models.py](#)

```
# [...]
from django.db import models
from django.core.urlresolvers import reverse

class Category(models.Model):
    # [...]
    def get_absolute_url(self):
        return reverse('astrowebapp.views.show_category', args=[self.alias])

class Project(models.Model):
    # [...]
    def get_absolute_url(self):
        return reverse('astrowebapp.views.show_project', args=[self.cat.alias, self.alias])
```

Configuration des urls du project coastdb

■ Module *urls* : [coastdb/urls.py](#)

```
from django.conf.urls import url
from django.contrib import admin
from astrowebapp import views

urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^app/(?P<category_alias>\w+)/$', views.show_category, name='category'),
    url(r'^app/(?P<category_alias>\w+)/(?P<project_alias>\w+)/$', views.show_project, name='project'),
]
```





Gabarits des vues

■ Affichage d'une catégorie et listing de ses projets : [astrowebapp/templates/astrowebapp/category.html](#)

► Template Django hérité de [astrowebapp/base.html](#).

```
{% extends "astrowebapp/base.html" %}
{% block title %}{% category.cat_name %}{% endblock title %}
{% block content %}
    <div class="col-xs-12"><div id="content">
        {{ category.summary|safe }}

        {% if proj_list %}<!-- Display list of published projects -->
        <h2>Available projects:</h2>
        <ul>
            {% for proj in proj_list %}
                <li><a href="{% url 'project' category.alias proj.alias %}">{{ proj.name }}</a></li>
            {% endfor %}
        </ul>
        {% endif %}
    </div></div>
{% endblock content %}
```

Définition des vues

■ Accès aux données d'une *Category* : [astrowebapp/views.py](#)

```
from django.shortcuts import render, get_object_or_404
from django.db import models

from .models import Project, Category

def show_category(request, category_alias):
    category = get_object_or_404(Category, alias=category_alias)
    context = {'category': category,
              'proj_list': Project.objects.filter(cat=category)}
    return render(request, 'astrowebapp/category.html', context)
```





Vue *category*

■ Affichage d'une catégorie et listing de ses projets

► http://127.0.0.1:8000/app/STAR_FORM

Star formation

The CoastDB database contains results of heavy numerical simulations computed in order to study the problem of star formation, essentially molecular cloud formation, evolution and collapse.

Understanding the dynamical evolution of the interstellar medium (ISM) and its relation to stellar birth is a key challenge in astronomy and astrophysics. The CoastDB project aims at providing observers and theorists studying formation and evolution of molecular clouds, their morphological and kinematical characteristics, and the formation of stars in their interior with a set of theoretical tools and a database of models to aid in the analysis and interpretation of current and future observations.

The goal of this database is to give access to observers, or more generally to any scientist working on a related field, to the results of these numerical simulations, which could be useful to help prepare or analyze observations.

Available projects:

- [Colliding flow simulations](#)





Django : liens utiles

■ La documentation de Django 1.10

▶ <https://docs.djangoproject.com/fr/1.10/>

■ Application django-suit (interface d'administration)

▶ <http://django-suit.readthedocs.io/en/develop/>

Autres applications utilisées (parmi 2602)

■ Application django-wysiwyg-redactor (éditeur de texte Imperavi)

▶ <https://github.com/douglasmiranda/django-wysiwyg-redactor>

■ Application django-autocomplete-light (autocomplétion)

▶ <http://django-autocomplete-light.readthedocs.io/en/master/>



2. Horus : Une IHM en PyQt4 pour la simulation numérique en astrophysique



Présentation

IHM de post-traitement de simulations en astrophysique :

- ▶ Simulations RAMSES (fluides auto-gravitants),
- ▶ Simulations ASH (MHD solaire),

Modèle de données basé sur SimDM :

- ▶ Projet
 - Simulation
 - Résultats
 - Produits

Implémentation en Python à l'aide du paquet PyQt4 (<https://riverbankcomputing.com/software/pyqt/>).





QtDesigner

■ Dessin de l'IHM en mode graphique : fichiers *.ui

The screenshot displays the Qt Designer environment. The main window is titled '- project_page.ui' and contains a form with the following elements:

- Buttons: '+ Add simulation code', '+ Add post-processing code', '+ Add target object', '+ Add simulation'
- Category: [Dropdown menu]
- Name: [Text input field]
- Directory: **CoastDB**
- Summary: [Text area]

The 'Inspecteur d'objet' (Object Inspector) on the right shows the following hierarchy:

- ProjectPage (QWin)
 - horizontalLayout_3 (QHB)
 - horizontalLayout_2 (QHB)
 - add_ppcode_pushButton (QPU)
 - add_simu_pushButton (QPU)
 - add_simucode_pushButton (QPU)
 - add_targobj_pushButton (QPU)
 - horizontalSpacer_buttons (Spacer)
 - header_label (QLat)
 - category_comboBox (QCo)
 - dir_alias (QLat)
 - dir_widget (Dir...)
 - groupBox (QGr)
 - alias_url_CoastDB_widget (Alia.)
 - horizontalSpacer_2 (Spacer)
 - groupBox_summary (QGr)
 - web_editor_widget (Web)
 - horizontalSpacer (Spacer)
 - horizontalSpacer_3 (Spacer)
 - label (QLat)
 - label_cat (QLat)
 - name_lineEdit (QLin)

The 'Éditeur de propriétés' (Property Editor) shows the properties for the selected 'web_editor_widget : WebEditorWidget':

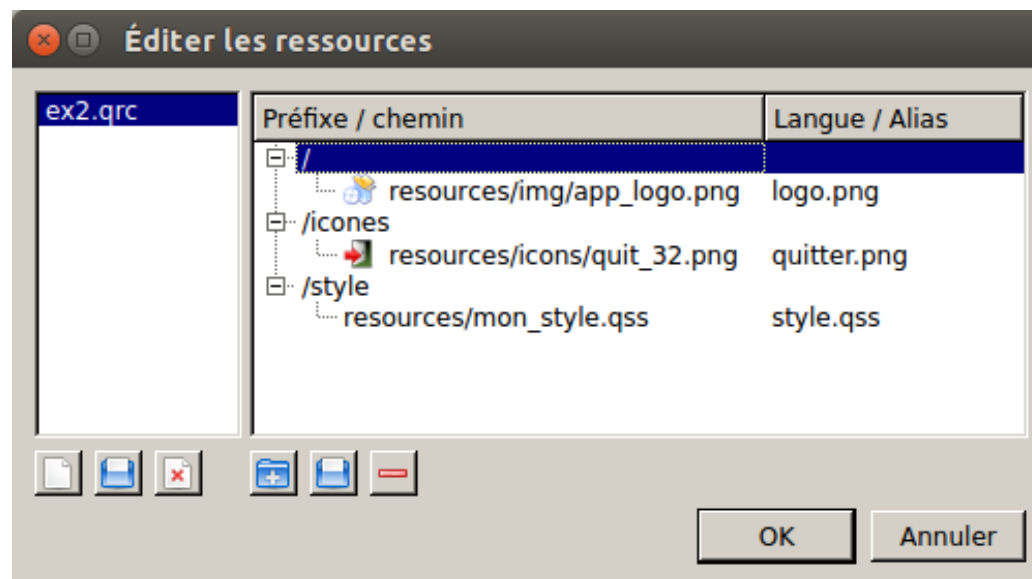
- QObject
 - objectName: wet
- QWidget
 - enabled: [checked]





QtDesigner

- Dessin de l'IHM en mode graphique : fichiers *.ui
- Édition de ressources (images, feuilles de style, icônes) : fichiers *.qrc





Outils de conversion PyQt4

- Conversion des *.qrc en modules Python : pyrcc4

```
> pyrcc4 horus.qrc -o horus_rc.py
```

- Conversion des *.ui en modules Python : pyuic4

```
> pyuic4 project_page.ui -o ui_project_page.py
```

▶ ui_project_page.py

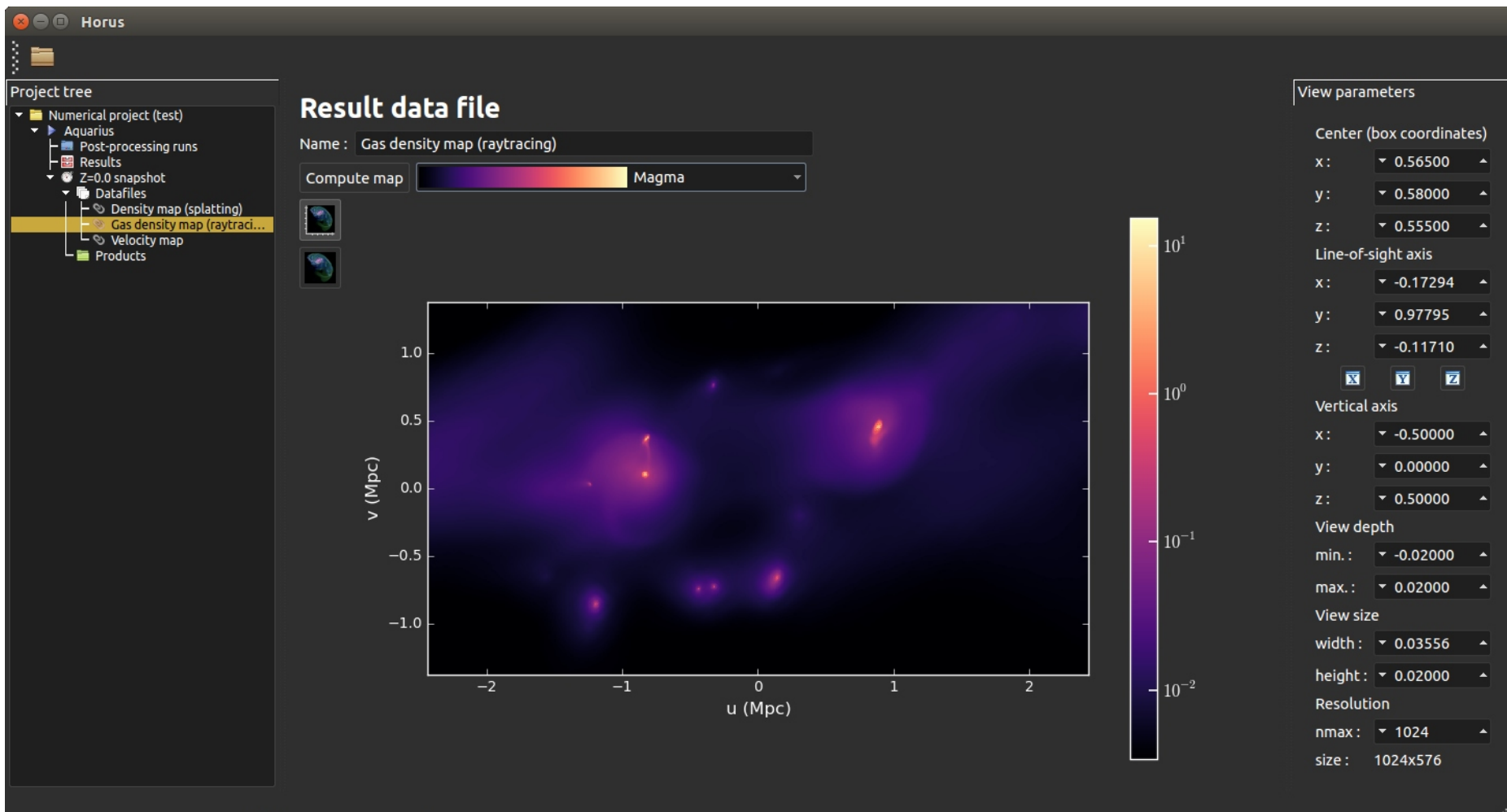
```
# -*- coding: utf-8 -*-
# Form implementation generated from reading ui file 'horus/ui/pages/project_page.ui'
#
# Created by: PyQt4 UI code generator 4.11.4
#
# WARNING! All changes made in this file will be lost!
from PyQt4 import QtCore, QtGui

# [...]

class Ui_ProjectPage(object):
    def SetupUi(self, ProjectPage):
        ProjectPage.setObjectName(_fromUtf8("ProjectPage"))
        self.gridLayout = QtGui.QGridLayout(ProjectPage)
        self.groupBox_summary = QtGui.QGroupBox(ProjectPage)
        self.verticalLayout = QtGui.QVBoxLayout(self.groupBox_summary)
        self.web_editor_widget = WebEditorWidget(self.groupBox_summary)
        self.verticalLayout.addWidget(self.web_editor_widget)
        self.gridLayout.addWidget(self.groupBox_summary, 6, 0, 1, 4)
        # [...]

from horus.ui.widgets.web_editor import WebEditorWidget
import horus_rc
```







Export au format « archive CoastDB »

■ Interface d'administration de CoastDB

- Upload de l'archive venant d'Horus (*.tar.gz),

The screenshot shows the 'CoastDB administration' interface. The top navigation bar includes the title 'CoastDB administration', the date and time 'Wednesday, 8. June 2016 01:58', and user information 'Welcome, Patrick.' with links for 'Change password' and 'Log out'. A left sidebar contains a search bar and a menu with items: Home, Projects, Protocols, Experiments (highlighted), Simulations (sub-menu open), Post-processing runs, Results, Products, and Statistical data. The main content area shows a breadcrumb trail 'Home » Coastdb » Simulations » Import simulation' and a light blue message box: 'Please select a compressed archive file (*.tar, *.tar.gz, *.tar.bz2) to upload into the database.' Below this is the 'Simulation upload form' with two rows: 'Format:' with a dropdown menu set to 'TAR archive (*.tar, *.tar.gz, *.tar.bz2)' and 'File to import:' with a 'Parcourir...' button and the text 'Aquarius.tar.gz'. A large blue 'Upload and preview' button is positioned to the right of the form.





Export au format « archive CoastDB »

■ Interface d'administration de CoastDB

- ▶ Upload de l'archive venant d'Horus (*.tar.gz),
- ▶ Prévisualisation de l'import dans la base de données,

CoastDB administration 🕒 Wednesday, 8. June 2016 01:58 Welcome, Patrick. [Change password](#) | [Log out](#)

Home » Coastdb » Simulations » **Import simulation**

Check the import preview and click 'Confirm import' to save the changes in the database.

Import preview

- Simulation 'Hydrodynamical simulation (imported)' successfully imported.
 - Snapshot 'Snapshot #20 (t=6.52 Myr)' successfully imported.
 - Snapshot 'Snapshot #22 (t=8.3 Myr)' successfully imported.
 - Product 'Clump catalog' successfully imported.
 - ProductDataFile 'Velocity dispersion map' successfully imported.
 - ProductFile 'irfu.png' successfully imported.
 - ResultDataFile 'Density map' successfully imported.
 - ResultFile 'mw_density_00030.png' successfully imported.
 - ResultDataFile 'Column density map (XY)' successfully imported.
 - ResultFile '00022_col_dens_xy.png' successfully imported.

[Confirm import](#)





Export au format « archive CoastDB »

Interface d'administration de CoastDB

- ▶ Upload de l'archive venant d'Horus (*.tar.gz),
- ▶ Prévisualisation de l'import dans la base de données,
- ▶ Données et fichiers joints déployés dans CoastDB.

Home Topics About Search project/simulation Patrick HENNEBELLE

Summary
Parameters
Algorithms
Applied physics
> Snapshots
t=6.52 Myr
t=8.3 Myr
-> t=11.25 Myr

Snapshots

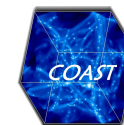
Snapshot #22 (t=8.3 Myr)

- Density map
- Column density map (XY)
- Column density map (YZ)
- Temperature cut (XY plane)
- Mass vs. density**
Mean mass per log. bins of density (in $M_{\odot} \cdot cm^{-3}$)
ASCII PNG
- Magnetic intensity vs. density
- Temperature vs. density





Une nouvelle base de données de simulation numérique pour COAST



■ Plate-forme collaborative pour la simulation numérique en astrophysique.

■ Horus : IHM de post-traitement (PyQt4) → CoastDB : application web. **django**

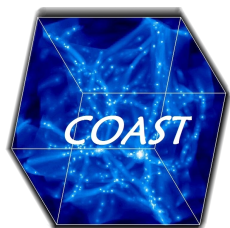
■ Phase de développement

- ▶ Release fin 2016

■ Travaux en cours

- ▶ Gestion de jobs de post-traitements avec PDL (CNRS-OBSPM) + intégration dans CoastDB,
- ▶ Outils de sélection/visualisation/extraction de données interactifs : composants web (D3JS, leaflets.js, ...),
- ▶ Plug-ins RAMSES/ASH/(autre ?) dans Horus.





Merci de votre attention...

Des questions ?

Commissariat à l'Énergie Atomique et aux Énergies Alternatives
C.E.A. Saclay | 91191 Gif-sur-Yvette Cedex | France
Tél. +33 (0)1 69 08 00 00

Etablissement public à caractère industriel et commercial | RCS Paris B 775 685 019

Damien CHAPON
Direction de la Recherche Fondamentale
CEA/DRF/IRFU/SEDI/LILAS
mail: damien.chapon@cea.fr
tel.: +33 (0)1 69 08 50 75