
GRID

Version 0.1

GRID Team

avr. 18, 2021

Table des matières

1	Table des matières	3
1.1	Introduction	3
1.2	Infrastructure	4
1.3	Indicateurs	4
1.4	Installation	4
1.5	Contacts	4
1.6	app	5
2	Index et recherche	9
	Index des modules Python	11
	Index	13

Petit texte de pres ici ?

1.1 Introduction

Ceci est un exemple

1.1.1 Titre 1

Sous-titre 1

Sous-sous titre 1

Code

```
a = 'hello';  
b = 'something';
```

In line code `result = a + 23`

Keyboard : Press `ctrl + s`

Boxes :

Astuce : Tip

Indication : Hint

Important : Important

Voir aussi :

See Also

Note : Note

Avertissement : Warning

Attention : Attention

Prudence : Caution

Erreur : Error

1.2 Infrastructure

1.2.1 Organisation global du répertoire

1.2.2 Point de vue Flask App / Python

1.2.3 Point de vue front-end / HTML

1.3 Indicateurs

1.3.1 Principe général de calcul

1.3.2 Exemple sur un indicateur

1.4 Installation

1.4.1 Installation en local

1.4.2 Déploiement en ligne

1.5 Contacts

Pour toute questions

1.6 app

1.6.1 app package

Subpackages

app.base package

Submodules

app.base.forms module

Copyright (c) 2019 - present AppSeed.us

```
class app.base.forms.CreateAccountForm(*args, **kwargs)
    Bases: flask_wtf.form.FlaskForm
    email = <UnboundField(TextField, ('Email',)), {'id': 'email_create', 'validators': [<wt
    password = <UnboundField(PasswordField, ('Password',)), {'id': 'pwd_create', 'validator
    username = <UnboundField(TextField, ('Username',)), {'id': 'username_create', 'validato

class app.base.forms.LoginForm(*args, **kwargs)
    Bases: flask_wtf.form.FlaskForm
    password = <UnboundField(PasswordField, ('Password',)), {'id': 'pwd_login', 'validators
    username = <UnboundField(TextField, ('Username',)), {'id': 'username_login', 'validato

class app.base.forms.Questionnaires(*args, **kwargs)
    Bases: flask_wtf.form.FlaskForm
    address = <UnboundField(TextField, ('Address',)), {'id': 'address', 'validators': [<wtf
    name_exploit = <UnboundField(TextField, ('Nom exploitation',)), {'id': 'exploit', 'vali
    submit = <UnboundField(SubmitField, ('Enregistrer',)), {}>
```

app.base.index_renderer module

```
app.base.index_renderer.plot_alerte()
app.base.index_renderer.plot_co2()
app.base.index_renderer.plot_ebitda()
app.base.index_renderer.plot_pct()
app.base.index_renderer.plots()
```

app.base.models module

Copyright (c) 2019 - present AppSeed.us

```
class app.base.models.User(**kwargs)
    Bases: sqlalchemy.ext.declarative.api.Model, flask_login.mixins.UserMixin
    email
    id
    password
```

username

```
app.base.models.request_loader(request)
app.base.models.user_loader(id)
```

app.base.routes module

Copyright (c) 2019 - present AppSeed.us

```
app.base.routes.access_forbidden(error)
app.base.routes.internal_error(error)
app.base.routes.login()
app.base.routes.logout()
app.base.routes.not_found_error(error)
app.base.routes.register()
app.base.routes.route_default()
app.base.routes.shutdown()
app.base.routes.unauthorized_handler()
```

app.base.util module

Copyright (c) 2019 - present AppSeed.us

```
app.base.util.hash_pass(password)
    Hash a password for storing.
app.base.util.verify_pass(provided_password, stored_password)
    Verify a stored password against one provided by user
```

app.base.zone_innondable module

Created on Thu Apr 15 16:02:54 2021

@author : Clement

```
class app.base.zone_innondable.DataImport(dpt, town, list_db)
    Bases: object
    create_folder(directory)
    import_data()
    quick_plot()
    read_json()
    to_json()

class app.base.zone_innondable.InternalPlot(geo_df)
    Bases: object
    plot()

app.base.zone_innondable.main(list_db, dpt, town)
app.base.zone_innondable.read_json(list_db, dpt, town)
```

Module contents

Copyright (c) 2019 - present AppSeed.us

app.home package

Submodules

app.home.routes module

Copyright (c) 2019 - present AppSeed.us

`app.home.routes.get_segment(request)`

`app.home.routes.index()`

`app.home.routes.route_template(template)`

Module contents

Copyright (c) 2019 - present AppSeed.us

Module contents

Copyright (c) 2019 - present AppSeed.us

`app.configure_database(app)`

`app.create_app(config)`

`app.register_blueprints(app)`

`app.register_extensions(app)`

CHAPITRE 2

Index et recherche

- genindex
- modindex
- search

a

- `app`, 7
- `app.base`, 7
- `app.base.forms`, 5
- `app.base.index_renderer`, 5
- `app.base.models`, 5
- `app.base.routes`, 6
- `app.base.util`, 6
- `app.base.zone_innondable`, 6
- `app.home`, 7
- `app.home.routes`, 7

A

`access_forbidden()` (dans le module `app.base.routes`), 6
`address` (attribut `app.base.forms.Questionnaires`), 5
`app`
 module, 7
`app.base`
 module, 7
`app.base.forms`
 module, 5
`app.base.index_renderer`
 module, 5
`app.base.models`
 module, 5
`app.base.routes`
 module, 6
`app.base.util`
 module, 6
`app.base.zone_innondable`
 module, 6
`app.home`
 module, 7
`app.home.routes`
 module, 7

C

`configure_database()` (dans le module `app`), 7
`create_app()` (dans le module `app`), 7
`create_folder()` (méthode `app.base.zone_innondable.DataImport`), 6
`CreateAccountForm` (classe dans `app.base.forms`), 5

D

`DataImport` (classe dans `app.base.zone_innondable`), 6

E

`email` (attribut `app.base.forms.CreateAccountForm`), 5

`email` (attribut `app.base.models.User`), 5

G

`get_segment()` (dans le module `app.home.routes`), 7

H

`hash_pass()` (dans le module `app.base.util`), 6

I

`id` (attribut `app.base.models.User`), 5
`import_data()` (méthode `app.base.zone_innondable.DataImport`), 6
`index()` (dans le module `app.home.routes`), 7
`internal_error()` (dans le module `app.base.routes`), 6
`InternalPlot` (classe dans `app.base.zone_innondable`), 6

L

`login()` (dans le module `app.base.routes`), 6
`LoginForm` (classe dans `app.base.forms`), 5
`logout()` (dans le module `app.base.routes`), 6

M

`main()` (dans le module `app.base.zone_innondable`), 6
module
 `app`, 7
 `app.base`, 7
 `app.base.forms`, 5
 `app.base.index_renderer`, 5
 `app.base.models`, 5
 `app.base.routes`, 6
 `app.base.util`, 6
 `app.base.zone_innondable`, 6
 `app.home`, 7
 `app.home.routes`, 7

N

`name_exploit` (attribut `app.base.forms.Questionnaires`), 5
`not_found_error()` (dans le module `app.base.routes`), 6

P

`password` (attribut `app.base.forms.CreateAccountForm`), 5
`password` (attribut `app.base.forms.LoginForm`), 5
`password` (attribut `app.base.models.User`), 5
`plot()` (méthode `app.base.zone_innondable.InternalPlot`), 6
`plot_alerte()` (dans le module `app.base.index_renderer`), 5
`plot_co2()` (dans le module `app.base.index_renderer`), 5
`plot_ebitda()` (dans le module `app.base.index_renderer`), 5
`plot_pct()` (dans le module `app.base.index_renderer`), 5
`plots()` (dans le module `app.base.index_renderer`), 5

Q

`Questionnaires` (classe dans `app.base.forms`), 5
`quick_plot()` (méthode `app.base.zone_innondable.DataImport`), 6

R

`read_json()` (dans le module `app.base.zone_innondable`), 6
`read_json()` (méthode `app.base.zone_innondable.DataImport`), 6
`register()` (dans le module `app.base.routes`), 6
`register_blueprints()` (dans le module `app`), 7
`register_extensions()` (dans le module `app`), 7
`request_loader()` (dans le module `app.base.models`), 6
`route_default()` (dans le module `app.base.routes`), 6
`route_template()` (dans le module `app.home.routes`), 7

S

`shutdown()` (dans le module `app.base.routes`), 6
`submit` (attribut `app.base.forms.Questionnaires`), 5

T

`to_json()` (méthode `app.base.zone_innondable.DataImport`), 6

U

`unauthorized_handler()` (dans le module `app.base.routes`), 6
`User` (classe dans `app.base.models`), 5
`user_loader()` (dans le module `app.base.models`), 6
`username` (attribut `app.base.forms.CreateAccountForm`), 5
`username` (attribut `app.base.forms.LoginForm`), 5
`username` (attribut `app.base.models.User`), 5

V

`verify_pass()` (dans le module `app.base.util`), 6