V.0.6.0 Changes

Generic URLs

Aurora v.0.6.0 supports the following convertors to generate generic URLs:

Type	Description
int	Accepts integers.
float	Accepts floating point values.
str	Accepts text without slashes (the default).
path	Accepts text with slashes.

```
Controller URL: greet/<name>
Controller URL: greet/<str:name>
Controller URL: edit/<int:id>
```

View Method

You don't need to provide`.html` in the View method anymore, only the view name is enough:

```
...
class Index(Controller):
    ...
    def get(self):
        return View(view="index")
...
```

Aurora v.0.6.0 added an optional the `app` argument, that you can use to set the app name explicitly:

```
...
class Index(Controller):
    ...
    def get(self):
        return View(view="index", app="notes")
...
```

You can also explicitly set the status code using the newly added `code` argument:

```
...
class Index(Controller):
    ...
    def get(self):
        return View(view="index", app="notes", code=302)
...
```

`302` is the default status code for the view.

If you wanted to return other app views, for whatever reason you can do it like:

```
...
class Index(Controller):
    ...
    def get(self):
        return View(view="another_view", app="another_app")
...
```

Common Statics

Aurora v.0.6.0 added a new directory called `assets` to statics with a few `.css` and `.js` files that you can use commonly for all of your child apps.

Error Method

`Error` method is no longer available, instead you can use the `View` method with a given `code` argument.

```
from aurora import View
...
return View(view="404", app="errors", code=404)
```

If you take a look at errors app controller you can see that a few changes occurred in the controllers structures.

Now you can manage `errors` app controllers using the CLI app. However, you should pass a non-numeric value for the Controller URL and then change it to a integer status code manually in the `_controllers.py` module.

Aurora Security

Aurora v.0.6.0 have added a module called security that you can take benefit from. This library provides a couple of methods that we have mentioned at the following:

Hashing Passwords

- `hash_password` - For hashing sensitive data (like passwords).
- `check_password` - For validating the hashed data with the requested data.

URL Redirecting

Using aurora security module, you can restrict URLs for different type of users. Aurora security provides the following methods for redirecting:

- `abort` – Redirects to an errors app controller via a given status code.
- `redirect` – Redirects to a URL using a given status code.
- `redirect-to` – Redirects to a controller using the app name and a controller name.
- `login_required` decorator – Redirects not logged-in users.
- `login_abort` decorator – Redirects logged-in users.

Validating Users

Aurora security module also provides the following methods to validate the users:

- `check_session` – Checks if a session exists.
- `get_session` - Returns a session value via a given name.
- `set_session` - Sets a session via a given name and value.
- `unset_session` - Unsets a session via a given name.
- `check_cookie` – Checks if a cookie exists.
- `get_cookie` - Returns a cookie value via a given name.
- `set_cookie` - Sets a cookie via a given name and value.
- `unset_cookie` - Unsets a cookie via a given name.