Investigación sobre Json y Web Services

Json

Python te permite usar formato intercambiable de datos popular llamado JSON (JavaScript Object Notation). El módulo estándar llamado **json** puede tomar datos de Python con una jerarquía, y convertirlo a representaciones de cadena de caracteres; este proceso es llamado *serializing*. El formato JSON es comúnmente usado por aplicaciones modernas para permitir intercambiar datos.

Si tienes un objeto x, puedes ver su representación JSON con una simple línea de código:

```
>>> import json
>>> json.dumps([1, 'simple', 'lista'])
'[1, "simple", "lista"]'
```

JSON (JavaScript Object Notation) es un formato sencillo para el intercambio de información. El formato JSON permite representar estructuras de datos (arrays) y objetos (arrays asociativos) en forma de texto. La notación de objetos mediante JSON es una de las características principales de JavaScript y es un mecanismo definido en los fundamentos básicos del lenguaje.

Para crear un array normal mediante JSON, se indican sus valores separados por comas y encerrados entre corchetes.

```
var modulos = ["Lector RSS", "Gestor email", "Agenda", "Buscador", "Enlaces"];
```

La notación JSON permite definir los arrays asociativos de una forma mucho más concisa.

```
var modulos = new Array();
modulos.titulos = {rss: "Lector RSS", email: "Gestor de email", agenda: "Agenda"};
```

La notación JSON para los arrays asociativos se compone de tres partes:

1. Los contenidos del array asociativo se encierran entre llaves ({ y })
2. Los elementos del array se separan mediante una coma (,)
3. La clave y el valor de cada elemento se separan mediante dos puntos (:)

Web Services

Configuraciones:

1. Install `django-summernote` to your python environment.

```
pip install django-summernote
```

1. Add `django_summernote` to `INSTALLED_APP` in `settings.py`.

```
INSTALLED_APPS += ('simple_webservice', )
```

1. Add `django_summernote.urls` to `urls.py`.

```
urlpatterns = patterns('',
    ...
    url(r'^ws/', include('simple_webservice.urls', namespace='ws')),
    ...
)
```

1. Add `webservice_autodiscover()` an the top os `urls.py`

```
from simple_webservice import webservice_autodiscover
webservice_autodiscover()

urlpatterns = patterns('',
    ...
)
```

Uso:

1. Crea un archivo *webservices.py* dentro de la aplicación con los servicios web
2. Dentro del archivo, copia y pega la siguiente línea:

```
import simple_webservice as ws
```

Bibliografía

Eguíluz Pérez, J. (2008). *Introducción a AJAX*.

Python Software Foundation. *Python*. Obtenido de https://pypi.python.org/pypi/django-simple-webservice/0.1

Python Software Foundation. *Tutorial de Python*. Obtenido de http://docs.python.org.ar/tutorial/3/inputoutput.html#guardar-datos-estructurados-con-json