

```
sudo apt-get install zlib1g-dev
```

```
1- pip install virtualenv
```

```
2- virtualenv my_env
```

```
3- cd my_env
```

```
4- source bin/activate
```

```
5- pip install Django
```

```
7- pip install lxml, pytz, spyne
```

```
from spyne.application import Application
```

```
from spyne.protocol.soap import Soap11
```

Application is the glue between one or more service definitions, interface and protocol choices.

```
from spyne.decorator import srpc
```

The @srpc decorator exposes methods as remote procedure calls and declares the data types it accepts and returns.

```
from spyne.service import ServiceBase
```

is the base class for all service definitions.

```
from spyne.model.primitive import String
```

```
from spyne.model.complex import Iterable
```

The names of the needed types for implementing this service should be self-explanatory.

```
class HelloWorldService(ServiceBase):
```

```
    @srpc(String, returns=String)
```

```
    def say_hello(name):
```

```
        return name
```

```
app = Application([HelloWorldService],
```

```
    'our.first.app',
```

```
    in_protocol=Soap11(validator='lxml'),
```

```
    out_protocol=Soap11(),
```

```
)
```

```
from django.conf.urls import patterns, include, url
```

```
from spyne.server.django import DjangoView
```

```
from app.views import app
```

```
urlpatterns = patterns("",
```

```
    url(r'^api/', DjangoView.as_view(application=app)),
```

)

Client

```
from suds.client import Client  
hello_client = Client('http://localhost:8000/?wsdl')  
print hello_client.service.say_hello("hello")
```