What is Web services?

- Web services are application components
- Web services communicate using open protocols
- Web services are self-contained and self-describing
- Web services can be used by other applications
- HTTP and XML is the basis for Web services

Benefits of Web Services

- Distributed component model
- Interoperability in a heterogeneous environment (no
- knowledge about partner)
- Communication through the firewall
- Saves development time because existing components can be exposed as Web services
- Freedom of programming choices
- Different communication message styles.

Web services Types

• **Conceptual level** (Cleint, server)
• **Technical level** (Big, RESTful)

Big Web services

Message Format: use XML messages that follow the Simple Object Access Protocol (SOAP) standard.
XML language defining a message architecture and message formats written in the Web Services Description Language (WSDL), an XML language for defining interfaces syntactically.

WSDL

A description language to define Web service interfaces, it answers 3 key questions about a Web service:

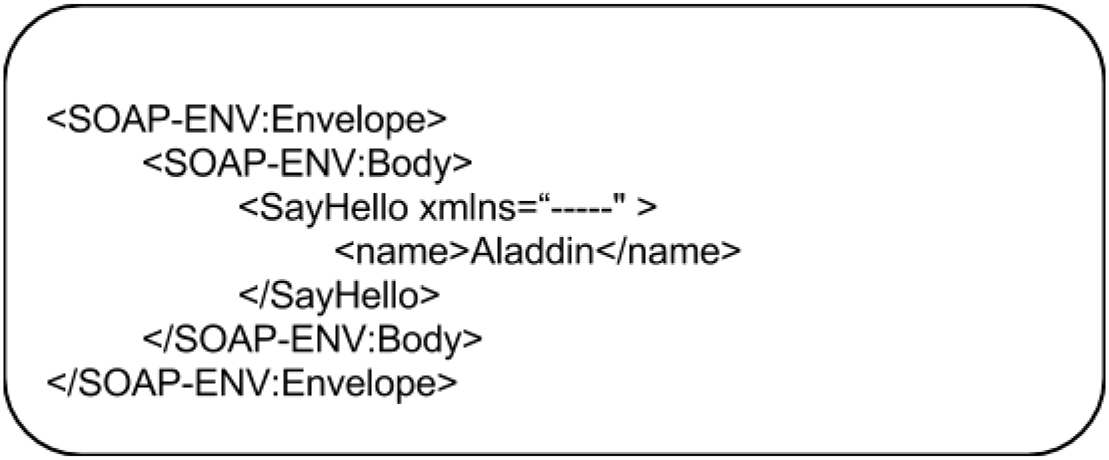
• Where a service is located?
• What a service offers?
• How a service is accessed?

WSDL file is a contract between web service and application program (Web Service Client).

SOAP

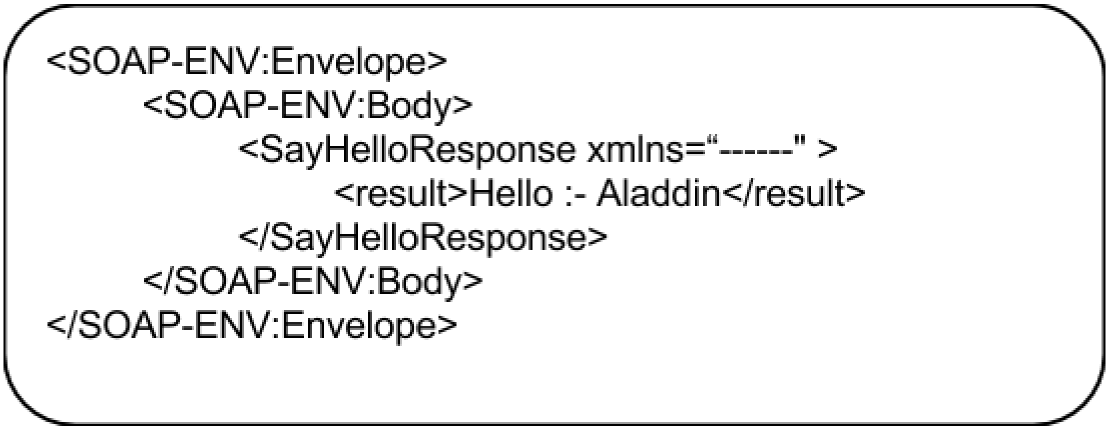
• Simple Object Access Protocol (SOAP) is a lightweight, XML-based protocol for exchanging information in a distributed environment.

SOAP Request



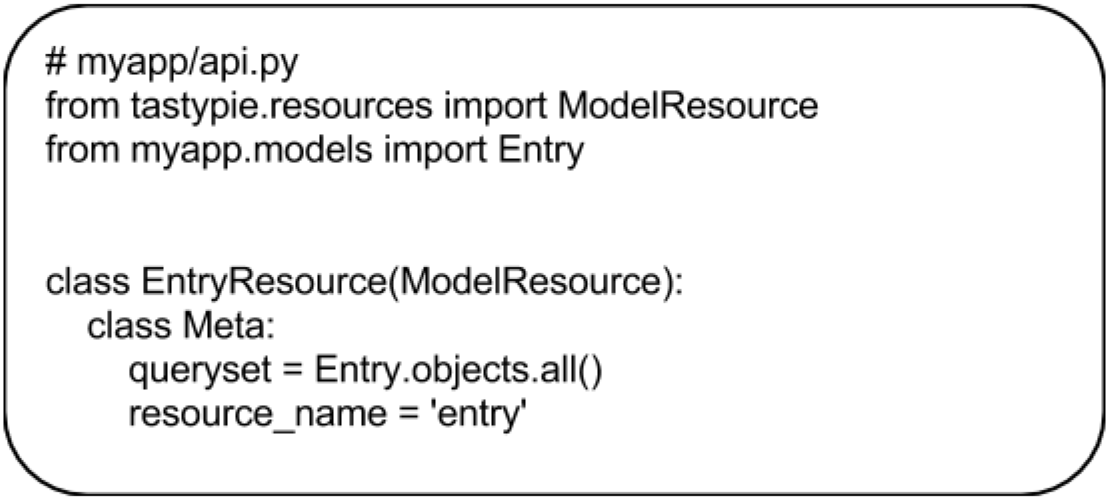
```
<SOAP-ENV:Envelope>
        <SOAP-ENV:Body>
                <SayHello xmlns=“-----" >
                        <name>Aladdin</name>
                </SayHello>
        </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP Response



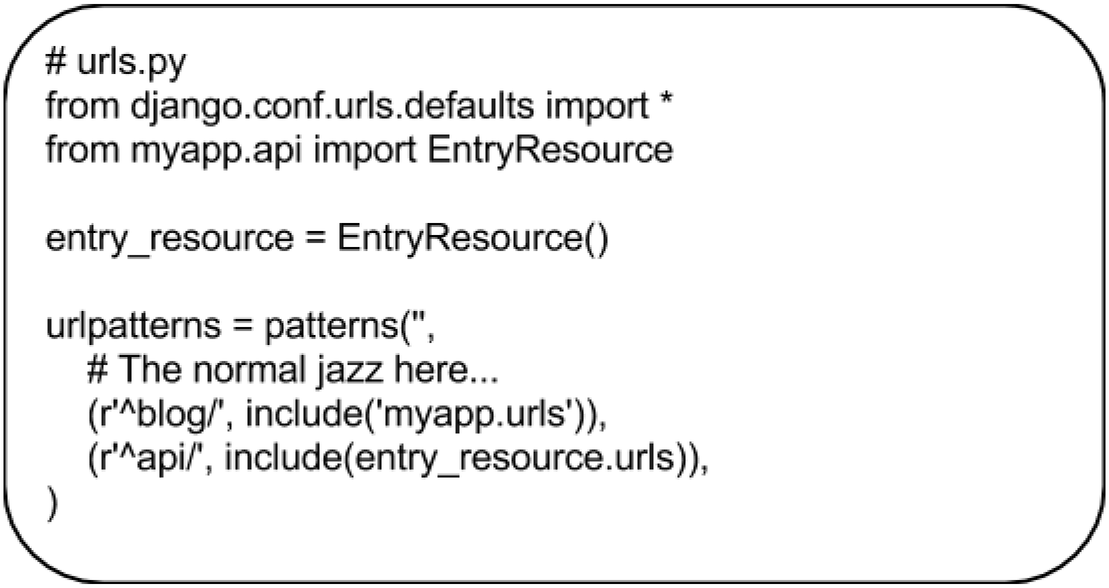
```
<SOAP-ENV:Envelope>
        <SOAP-ENV:Body>
                <SayHelloResponse xmlns=“------" >
                        <result>Hello :- Aladdin</result>
                </SayHelloResponse>
        </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Getting Started with Tastypie



```
# myapp/api.py
from tastypie.resources import ModelResource
from myapp.models import Entry


class EntryResource(ModelResource):
    class Meta:
        queryset = Entry.objects.all()
        resource_name = 'entry'
```



```
# urls.py
from django.conf.urls.defaults import *
from myapp.api import EntryResource

entry_resource = EntryResource()

urlpatterns = patterns('',
    # The normal jazz here...
    (r'^blog/', include('myapp.urls')),
    (r'^api/', include(entry_resource.urls)),
)
```

- http://127.0.0.1:8000/api/entry/?format=json
- http://127.0.0.1:8000/api/entry/1/?format=json
- http://127.0.0.1:8000/api/entry/schema/?format=json
- http://127.0.0.1:8000/api/entry/set/1;3/?format=json

fields

excludes