

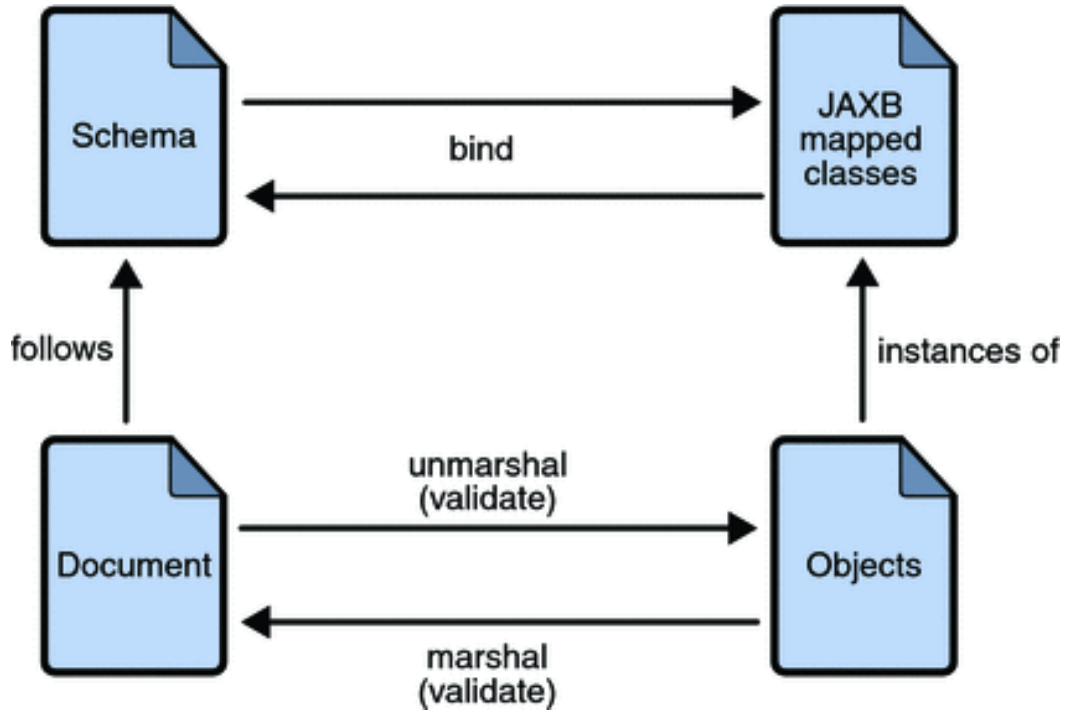


# SPRING WEB SERVICES

**SOAP WEB SERVICES**

MARCH, 2015

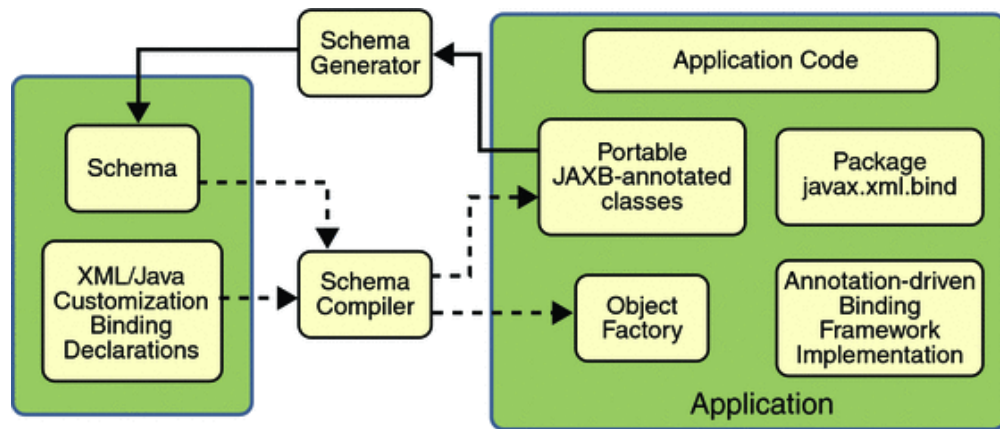
# AUTOMATIC OBJECTS MARSHALLING



# XML SCHEMA

```
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="tasks">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="employee" type="Employee"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="Employee">
    <xs:sequence>
      <xs:element name="id" type="xs:long"/>
      <xs:element name="fullName" type="xs:string"/>
      <xs:element name="title" type="xs:string"/>
      <xs:element name="tasks">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="task" minOccurs="1" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:sequence>
    </xs:complexType>
  </xs:schema>
```

# JAXB UTILITIES



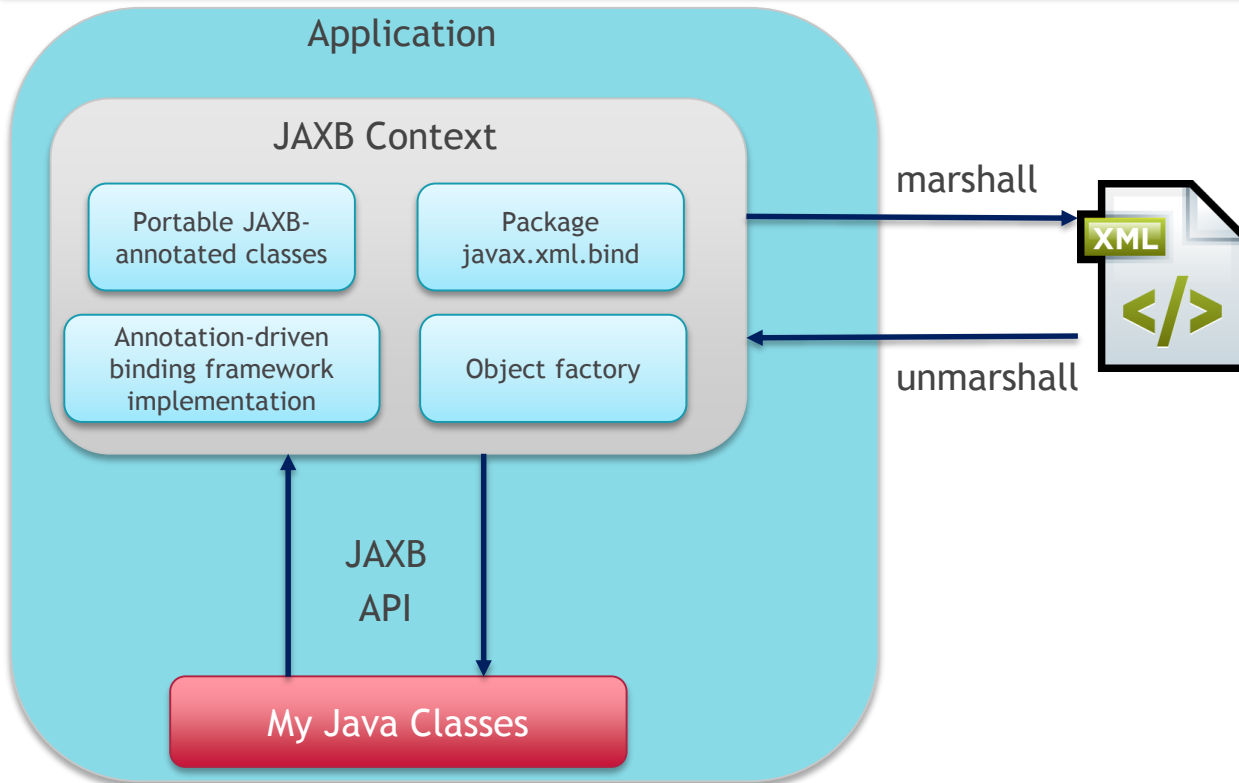
- Xjc - generate java from xsd/wsdl

```
xjc [-options ...] <schema file/URL/dir/jar>...
```

- Schemagen - generate xsd/wsdl from java pojo's

```
schemagen [-d path] [java-source-files]
```

# JAXB MARSHALLING



# SPRING-WS SERVICE ENDPOINT

@Endpoint

```
public class CountryEndpoint {  
    private static final String NAMESPACE_URI = "http://company.com/country-web-service";  
  
    private CountryRepository countryRepository;  
  
    @Autowired  
    public CountryEndpoint(CountryRepository countryRepository) {  
        this.countryRepository = countryRepository;  
    }  
  
    @PayloadRoot(namespace = NAMESPACE_URI, localPart = "getCountryRequest")  
    @ResponsePayload  
    public GetCountryResponse getCountry(@RequestPayload GetCountryRequest request) {  
        GetCountryResponse response = new GetCountryResponse();  
        response.setCountry(countryRepository.findCountry(request.getName()));  
  
        return response;  
    }  
}
```

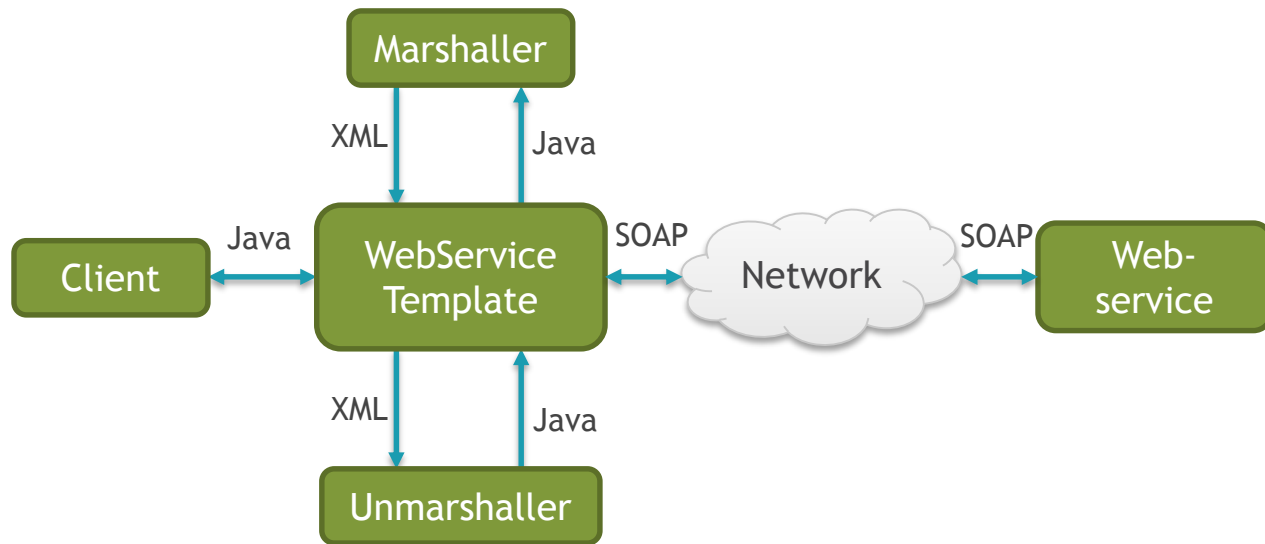
# SPRING-WS CONFIGURATION

@EnableWs

@Configuration

```
public class WebServiceConfig extends WsConfigurerAdapter {  
    @Bean  
    public ServletRegistrationBean messageDispatcherServlet(ApplicationContext applicationContext) {  
        MessageDispatcherServlet servlet = new MessageDispatcherServlet();  
        servlet.setApplicationContext(applicationContext);  
        servlet.setTransformWsdlLocations(true);  
        return new ServletRegistrationBean(servlet, "/ws/*");  
    }  
  
    @Bean(name = "countries")  
    public DefaultWsd11Definition defaultWsd11Definition(XsdSchema countriesSchema) {  
        DefaultWsd11Definition wsdl11Definition = new DefaultWsd11Definition();  
        wsdl11Definition.setPortTypeName("CountriesPort");  
        wsdl11Definition.setLocationUri("/ws");  
        wsdl11Definition.setTargetNamespace("http://company.com/country-web-service");  
        wsdl11Definition.setSchema(countriesSchema);  
        return wsdl11Definition;  
    }  
  
    @Bean  
    public XsdSchema countriesSchema() {  
        return new SimpleXsdSchema(new ClassPathResource("countries.xsd"));  
    }  
}
```

# SPRING-WS CLIENT SIDE





# SENDING REQUEST TO WEB SERVICE

```
public class WebServiceClient {

    private static final String MESSAGE =
        "<message xmlns='http://tempuri.org'>Hello Web Service World</message>";

    private final WebServiceTemplate webServiceTemplate = new WebServiceTemplate();

    // send to the configured default URI
    public void simpleSendAndReceive() {
        StreamSource source = new StreamSource(new StringReader(MESSAGE));
        StreamResult result = new StreamResult(System.out);
        webServiceTemplate.sendSourceAndReceiveToResult(source, result);
    }

    // send to an explicit URI
    public void customSendAndReceive() {
        StreamSource source = new StreamSource(new StringReader(MESSAGE));
        StreamResult result = new StreamResult(System.out);

        webServiceTemplate.sendSourceAndReceiveToResult("http://localhost:8080/AnotherWebService",
            source, result);
    }
}
```

# WEB SERVICE TEMPLATE CONFIGURATION

```
<bean id="messageFactory"
      class="org.springframework.ws.soap.saaj.SaajSoapMessageFactory"/>

<bean id="webServiceTemplate"
      class="o.s.ws.client.core.WebServiceTemplate">
  <constructor-arg ref="messageFactory"/>
  <property name="messageSender">
    <bean class="o.s.ws.transport.http.HttpComponentsMessageSender">
      <property name="credentials">
        <bean class="org.apache.http.auth.UsernamePasswordCredentials">
          <constructor-arg value="john:secret"/>
        </bean>
      </property>
    </bean>
  </property>
</bean>
</property>
<property name="defaultUri" value="http://example.com/WebService"/>
</bean>
```

# CLIENT SIDE JAVA CODE GENERATION



Xjc:

```
xjc -wsdl <wsdlurl> -d <directory>
```

Ant:

```
<taskdef name="xjc" classname="com.sun.tools.xjc.XJCTask">
  <classpath refid="classpathid" />
</taskdef>

<target name="generate.wsdl" description="Generate XML binding artifacts">
  <xjc destdir="${basedir}/src" package="generated.webClient.country" removeOldOutput="yes">
    <arg value="-verbose" />
    <arg value="-wsdl" />
    <arg value="${basedir}/conf/country.wsdl" />
  </xjc>
</target>
```

# WEB SERVICE REQUEST WITH MARSHALLING

```
public GetCityForecastByZIPResponse getCityForecastByZip(String zipCode) {  
    // GetCityForecastByZIP was generated from WSDL  
    GetCityForecastByZIP request = new GetCityForecastByZIP();  
    request.setZIP(zipCode);  
  
    SoapActionCallback callback =  
        new SoapActionCallback("http://company.com/WeatherWS/GetCityForecastByZIP");  
  
    // GetCityForecastByZIPResponse was generated from WSDL  
    GetCityForecastByZIPResponse response =  
        (GetCityForecastByZIPResponse) getWebServiceTemplate()  
            .marshalSendAndReceive(request, callback);  
  
    return response;  
}
```



# THANK YOU!

**MAKSYM\_GOVORISCHEV@EPAM.COM**

MARCH, 2015