



```
@Path("/api")
class Api {
  @POST
  @Consumes({"application/json", "application/xml"})
  @Produces({"application/json", "application/xml"})
  public byte[] resourceMethod(InputStream
ins,@HeaderParam("Content-Type") String contentType,
@HeaderParam("Accept") String accept) {
    if(contentType.eq("json")) {
      // convert json data into object format
    }else if(contentType.eq("xml") {
      // convert xml data into object format
    }

    // use this object of data to perform business logic
    and produce the another object as an output

    if(accept.eq("json")) {
      // convert object computed in business logic into
      json format response
    }else if(accept.eq("xml")) {
      // convert object computed in business logic into
      xml format response
    }
    return response;
  }
}
```

#build the business logic to work around objects

```
@Path("/api")
class Api {
  @POST
  @Consumes(..)
  @Produces(..)
  public byte[] somemethod(InputStream ins) {
    // convert the raw data into object

    // use the object to perform business
    logic and compute object of data as an output

    // convert the object computed into raw
    format, to dispatch to client

    return response;
  }
}
```

duplicated across  
several resource classes

```
@Path("/api")
class Api {
  @POST
  @Consumes(..)
  @Produces(..)
  public byte[] somemethod(InputStream ins) {
    // invoke the message converter to
    convert raw data into object

    // use the object to perform business
    logic and compute object of data as an output

    // invoke the message converter to
    convert object into raw format

    return response;
  }
}
```

```
class MessageConverter {
  toObject() {}
  fromObject() {}
}
```

```
@Path("/api")
class Api {
  @POST
  @Consumes(..)
  @Produces(..)
  public Object somemethod(Object obj) {
    // business logic
    return object;
  }
}
```