Data Transfer Object

- **Bean**
- **Information Hiding**
  - Remove Sensitive Data
    - Account Number
    - IDs
- **Reduction of Network Traffic**
  - Complex Data Structures
- **Standard Communication**
- **Openness**
- **Text Based Data Formats**
  - XML
  - JSON
- **Platform Independent**

**Server**
- Java, C#, PHP
- REST API
- SOAP

**Client**
- HTML, JavaScript, AJAX
- Web Service
  - Service-to-Service
- Desktop Application
DTO vs POJO

**DTO**
- Data Structure
- Bean
- Complex
- Marshalling

**POJO**
- OOP
- Information Hiding
- Inheritance
- Business Logic
Data Transfer Objects

Request/Response Objects

- Different Data Structures
  - Requests / Responses
  - per Recording Service
  - Flexibility
  - Mandatory and Optional Fields

- Bean
  - Default Constructor
  - Public getter/setter Methods
  - In Controller Only!

- Data Formats
  - Binary (server side)
  - XML (schema validation)
  - JSON (compactness)

Challenges

- Automatic Conversion
  - Marshalling
    - Object → Text
    - Text → Object
- Multiple Formats
- Transformations
  - JSON ↔ XML
- Schema Definition / Validation
Generation

- **XJC**
  - tool, application
  - XSD → Annotated Java Class

- **Maven**
  - jaxb2-maven-plugin
  - Tags
    - packageName
    - sources
  - Location /target/generated-sources
  - Ignore Version Control!

- XSD Complex Types
- Schema Validation
- Data Types
- Generated Beans
  - XML Mapping
  - JSON Mapping
- jackson Parser
Table of Contents

1. Data Transfer Objects
2. REST API Documentation
3. WSDL
Types of Web Services

REST API
- Simple
- JSON Messages
  - Currently popular

Web Service
- WSDL
- XML
- SOAP
Recommended Structure

- Title
- URL Pattern
- Method
- URL Parameters
- Body Parameters
- Successful Response
- Error Response
- Request example
- Notes
Title

1. List Measurements
   - type: Query / Fetch
   - purpose: clear
   - count: many records

2. Record Measurements
   - type: recording
   - purpose: understandable
   - count: single record

3. Query Measurement
   - number of record is not defined

- Short
- Descriptive
- Define the service
  - Type
  - Purpose
  - Count
- Sort description is optional
URL Pattern

- Location of the service
- Path only
- Unnecessary
  - Server
  - Port
  - Application
- URL Mapping – `web.xml`
- Parameters
- `/measurements`
- `/measurements/::<id>`
Method

- HTTP Method
- Single URL – Multiple Methods
- Single URL – Multiple Services
- Different Method – Different Behavior

/ measurement/:id

**GET** Query a single record by id.

**POST** Record of modify a record with the given id.
URL Parameters

- GET Method
- Query Parameters
- Key–Value Pairs
- Mandatory or Optional
- Expected Value
  - Number
  - Text
  - Date (Format)
- URL Encoding
Body Parameters

- POST Method
- JSON Structure Definition
- Data Structure
- Restriction
- Data Transfer Object
Successful Response

- Status Code
- Response Body
  - HTML
  - JSON
- Structure description (JSON)
Error Response

- Many different errors
- Different error, different status codes
- Using existing HTTP Status Codes
- Additional meaning
- Error Message Structures
- JSON
- Typical Errors
  - Permission
  - Unexpected Parameters
  - Service unavailable
  - Validation or Sanitize Error
Example Request

- Executable format
  - curl
  - AJAX
- Facilitate the usage
- Easy to understand
Notes

- Explanation of ambiguous parts.
- Faults and fixes documentation.
- Versioning

Suggestion

- Timestamp
- Author
- Note
Table of Contents

1. Data Transfer Objects
2. REST API Documentation
3. WSDL
<table>
<thead>
<tr>
<th>W3C 2007</th>
</tr>
</thead>
<tbody>
<tr>
<td>XML</td>
</tr>
<tr>
<td>XSD</td>
</tr>
<tr>
<td>Platform Independent</td>
</tr>
<tr>
<td>Text Based</td>
</tr>
<tr>
<td>Semi–Structured</td>
</tr>
<tr>
<td>Automatic Processing</td>
</tr>
<tr>
<td>Code Generation</td>
</tr>
</tbody>
</table>

Service
- Abstract
  - Types
  - Messages
  - Ports
- Implementation
  - URL, port
  - Class
Web Service Description Language

- Web service interface definition
  - Port type: Operation signatures
  - Messages: Parameter definitions
  - Types: Complex type definitions
  - Bindings: Transport protocol and payload format
- Web service implementation
  - Service: Service definition element
  - Ports: Supported interface bindings
<definitions>
<types>definition of types...</types>

<message>definition of a message...</message>

<portType>
<operation>definition of a operation...</operation>
</portType>

<binding>definition of a binding...</binding>

<service>definition of a service....</service>
</definitions>
Type Definition

- **types** tag
- XML Schema Definition
- Simple Types
  - Text
  - Number
  - true, false
- Restrictions
  - RegEx
  - Enumeration
  - Interval
- Complex Types
  - Arrays, Lists
  - Objects

```xml
<types>
  <schema
targetNamespace = "..."
xmlns = "...">
    <element name = "...">
      <complexType>
        ...
      </complexType>
    </element>
  </schema>
</types>
```
Message Definition

- Request/Response Messages
- Name
- Complex Structure
- Data Transfer Object
- Types
  - input
  - output
- Messages Parts

```
<message name = "SayHelloResponse">
  <part name = "greeting"
       type = "xsd:string"/>
</message>
```
Port Definition

- End Point of the Web Service
- Name
- Type
  - One-way
  - Request–Response
  - Solicit–Response
  - Notification
- Messages
  - Order
  - input
  - output
  - fault
One-Way, Request-Response

One-Way
- Service waits for Request
- No Response
- Only **input**
- HTTP 202 OK Response

Request-Response
- Clint sends Request and wait for Response
- **input and output**
- HTTP
Solicit-Response, Notification

Solicit-Response
- Server sends data
- Client sends Ack.
- Inverse behavior
- output and input definition

Notification
- Server sends data
- Client does not send Ack
- Only output
- Broadcasting
Binding Definition

- Definition of the Implementation
- `portType`
- Data Transfer
  - GET, POST
  - Simple Object Access Protocol
- Single `portType` – multiple binding
- binding
  - name
  - port
Service Definition

- End Point Definition
- **Assign** binding to Class
- Single Service may have multiple **port**.
- Where is the service?
- Via which port can it be accessed?
- Messages?
- Concrete URL