

## SOAP web services after Rosetta 7.1 SP

- **Product:** Rosetta
  - **Product Version:** 7.1
  - **Relevant for Installation Type:** Local
- 

### Description

Rosetta 7.1 Service Pack comes with an update to Java 15 (OpenJDK). As part of this update Rosetta also got a new Apache CXF version.

As a result, SOAP web services [wsdl](#) was modified.

The following APIs are affected: ProducerWebServices, SipWebServices and DepositWebServices

This may cause an external application SOAP interface to break.

### Resolution

1. It is advised to test Rosetta 7.1 SP on Sandbox environment prior to Production.
2. If existing SOAP API breaks, the application developer will have to rebuild the stub from the new wsdl.

wsdl is available on your environment under dpsws path as: <https://rosetta.exlibrisgroup.com/dpsws/>

3. consider new REST APIs - <https://rosetta.exlibrisgroup.com/rest/>

### Example

In the affected \*WebService interface class, change the name attribute of the @WebResult annotation to be camelCase:

From

```
@WebMethod
@WebResult(name = "CreateContact", targetNamespace = "")
@RequestWrapper(localName = "createContact", targetNamespace =
"http://dps.exlibris.com/", className = "com.exlibris.dps.CreateContact")
@ResponseWrapper(localName = "createContactResponse", targetNamespace =
"http://dps.exlibris.com/", className = "com.exlibris.dps.CreateContactResponse")
public String createContact(
    @WebParam(name = "arg0", targetNamespace = "")
    String arg0,
    @WebParam(name = "arg1", targetNamespace = "")
```

```
String arg1);
```

To

```
@WebMethod
@WebResult(name = "createContact", targetNamespace = "")
@RequestWrapper(localName = "createContact", targetNamespace =
"http://dps.exlibris.com/", className = "com.exlibris.dps.CreateContact")
@ResponseWrapper(localName = "createContactResponse", targetNamespace =
"http://dps.exlibris.com/", className = "com.exlibris.dps.CreateContactResponse")
public String createContact(
    @WebParam(name = "arg0", targetNamespace = "")
    String arg0,
    @WebParam(name = "arg1", targetNamespace = "")
    String arg1);
```

And in the corresponding \*Response class, remove the XmlElement annotation line.(Or fix the name attribute to be camelCase also):

```
@XmlAccessorType(XmlAccessType.FIELD)
@XmlType(name = "createContactResponse", propOrder = {"createContact"})
public class CreateContactResponse {

    @XmlElement(name = "CreateContact")
    protected String createContact;

    // Rest of code

}
```

- 
- Article last edited: 22-Nov-2021