

# JPOX Extension

## Introduction

JPOX-Spatial allows the use of [JPOX](#) as persistence layer for geospatial applications in an environment that supports the OGC SFA specification. It defines type mappings to let JPOX know how to persist a number of different Java geometry types (JTS, PostGIS-JDBC geometry types, Oracle's JGeometry). Thus a Java developer can mix spatial and non-spatial datatypes in his POJOs and can rely on JPOX to persist and query those types.

The mission for the JPOX Extension is to:

- Use of OGC Filter to produce a query against JPOX
- Expression evaluation against a POJO using XPath
- Glue code for JPOX-Spatial Domains Models to be viewed as a simple POJODataStore
- Additional utility code to assist intergration between POJO domain model and GeoTools as required

How you can help?

- provide a test data set (and public database) with an interesting "complex" domain model

The following needs to be sorted out:

- [#Filter Support](#)
- [#Data Access](#)
- [#Rendering](#)

The following documentation is available (or at least intended):

## Filter Support

Since most everything in the spatial world (pun intended) with OGC Filter - this is the first step. It will enable our Data Access and Rendering intergration.

## Pojo Property Accessor

We need to make use of the [Expression Improvements](#) PropertyAccessor API to teach the filter system about working with Plain-Old-Java-Objects.

This can be done in two parts:

- simple reflection to access Bean properties (or reflection straight to Java fields?)
- full xpath support with XPath (easy since beans + collections already supported)

This will allow the post processing filters to work.

## Filter to JPOX Query Transformation

This work has been done for BBox query; we will need to go through the list.

## Data Access

See the separate wiki page for this: [Dry Run at DataAccess Story](#)

# Rendering

Given the above two the following is needed for rendering intergration:

- Teach PropertyAccessor the concept of "default geometry"
  - Add getDefaultGeometry() to PropertyAccessor interface.
- Teach StreamingRenderer how to work with Content class rather then FeatureSource.
  - ok
- Implement DataAccess and Source for JPOX
  - ok