# Contents



# 1 CouchDB Database Investigation - Source Field Validation

## 1.1 Summary

Investigated CouchDB database `skol_taxa_dev` for integer values in `source` fields that could cause the PySpark schema type error.

**Result**: ✅ **All 5,278 documents have clean source fields** - no integers found.

## 1.2 Investigation Details

### 1.2.1 Database Checked

- **Database**: `skol_taxa_dev`
- **Server**: http://127.0.0.1:5984
- **Total Documents**: 5,278
- **Documents Checked**: 5,278 (excluding _design docs)

### 1.2.2 What Was Checked

For each document in the database: 1. Check if `source` field exists 2. If `source` is a dict, check each key-value pair 3. Flag any value that is

an integer instead of string/None

### 1.2.3 Code Used

```
import couchdb

# Connect to CouchDB
server = couchdb.Server('http://127.0.0.1:5984')
server.resource.credentials = ('admin', 'SU2orange!')
db = server['skol_taxa_dev']

# Check all documents
issues_found = []
for doc_id in db:
    if doc_id.startswith('_design'):
        continue
    doc = db[doc_id]
    if 'source' in doc and isinstance(doc['source'], dict):
        for key, value in doc['source'].items():
            if isinstance(value, int):
                issues_found.append({
                    'doc_id': doc_id,
                    'key': key,
                    'value': value
                })
```

### 1.2.4 Results

```
✓ Checked 5278 documents

✓ No integer values found in 'source' fields
  All source field values are strings or None as expected
```

## 1.3 Expected Source Field Structure

According to the schema, `source` should be:

```
StructField("source", MapType(StringType(), StringType(), valueContainsNull=True
```

This means: - **Keys**: StringType (non-null) - **Values**: StringType or NULL - **Field**: Non-nullable (source dict must exist)

### 1.3.1 Valid Examples

```
# Valid
source = {
```

```
    'doc_id': 'abc123',
    'url': 'https://example.com/paper.pdf',
    'db_name': 'skol_dev'
}

# Also valid (None values OK)
source = {
    'doc_id': 'abc123',
    'url': None,  # valueContainsNull=True
    'db_name': 'skol_dev'
}
```

#### 1.3.2 Invalid Examples

```
# Invalid - integer value
source = {
    'doc_id': 'abc123',
    'url': None,
    'db_name': 'skol_dev',
    'paragraph_number': 68  # ← WRONG! Integer in source
}
```

### 1.4 Implications

Since all CouchDB documents have clean source fields, the error `MapType(StringType(), StringType(), True) can not accept object 68 in type int` must be occurring during:

1. **Extraction process**: When `extract_taxa(annotated_df)` converts Taxon objects to Rows
2. **Data transformation**: Some code path adds integers to the source dict
3. **PySpark version**: Different PySpark versions may validate differently

### 1.5 Next Steps

#### 1.5.1 If Error Persists in Notebook

Add debug validation to extract_taxa_to_couchdb.py:97-125:

```
def convert_taxa_to_rows(partition: Iterator[Taxon]) -> Iterator[Row]:
    for taxon in partition:
        taxon_dict = taxon.as_row()

        # DEBUG: Validate source dict
```

```
        if 'source' in taxon_dict and isinstance(taxon_dict['source'], dict):
            for key, value in taxon_dict['source'].items():
                if value is not None and not isinstance(value, str):
                    print(f"⚠️  Non-string in source['{key}']: {value} ({type(va
                    print(f"   Taxon: {taxon_dict.get('taxon', 'N/A')}")
                    taxon_dict['source'][key] = str(value)  # Convert to string

        if '_id' not in taxon_dict:
            taxon_dict['_id'] = generate_taxon_doc_id(
                taxon_dict['source']['doc_id'],
                taxon_dict['source'].get('url'),
                taxon_dict['line_number'] or 0
            )
        if 'json_annotated' not in taxon_dict:
            taxon_dict['json_annotated'] = None

        yield Row(**taxon_dict)
```

This will: 1. Detect the exact taxon causing the issue 2. Show what field has an integer value 3. Automatically convert it to string 4. Allow the extraction to continue

#### 1.5.2 Alternative: Check PySpark Version

```
import pyspark
print(f"PySpark version: {pyspark.__version__}")
```

Different versions may have different schema validation strictness.

## 1.6 Related Documentation

- SCHEMA_TYPE_FIX.md - Full analysis of schema type error
- TAXA_ID_JOIN_FIX.md - Recent refactoring adding _id field
- CIRCULAR_IMPORT_CHECK.md - Import validation

## 1.7 Files Involved

- extract_taxa_to_couchdb.py - Schema and extraction logic
- taxon.py - Taxon.as_row() method
- jupyter/ist769_skol.ipynb - Where error occurs

## 1.8 Conclusion

✅ **CouchDB data is clean** - All 5,278 documents have proper source field structure ✅ **Schema is correct** - MapType(StringType(),

StringType()) properly defined ✅ **Code fixes applied** - empirical_page_number properly converted to string ⚠️ **Error source unknown** - Must be in extraction process, add debug code to investigate

The investigation confirms that the stored data is not the source of the problem. The issue likely occurs during the extraction process when converting Taxon objects to PySpark Rows.