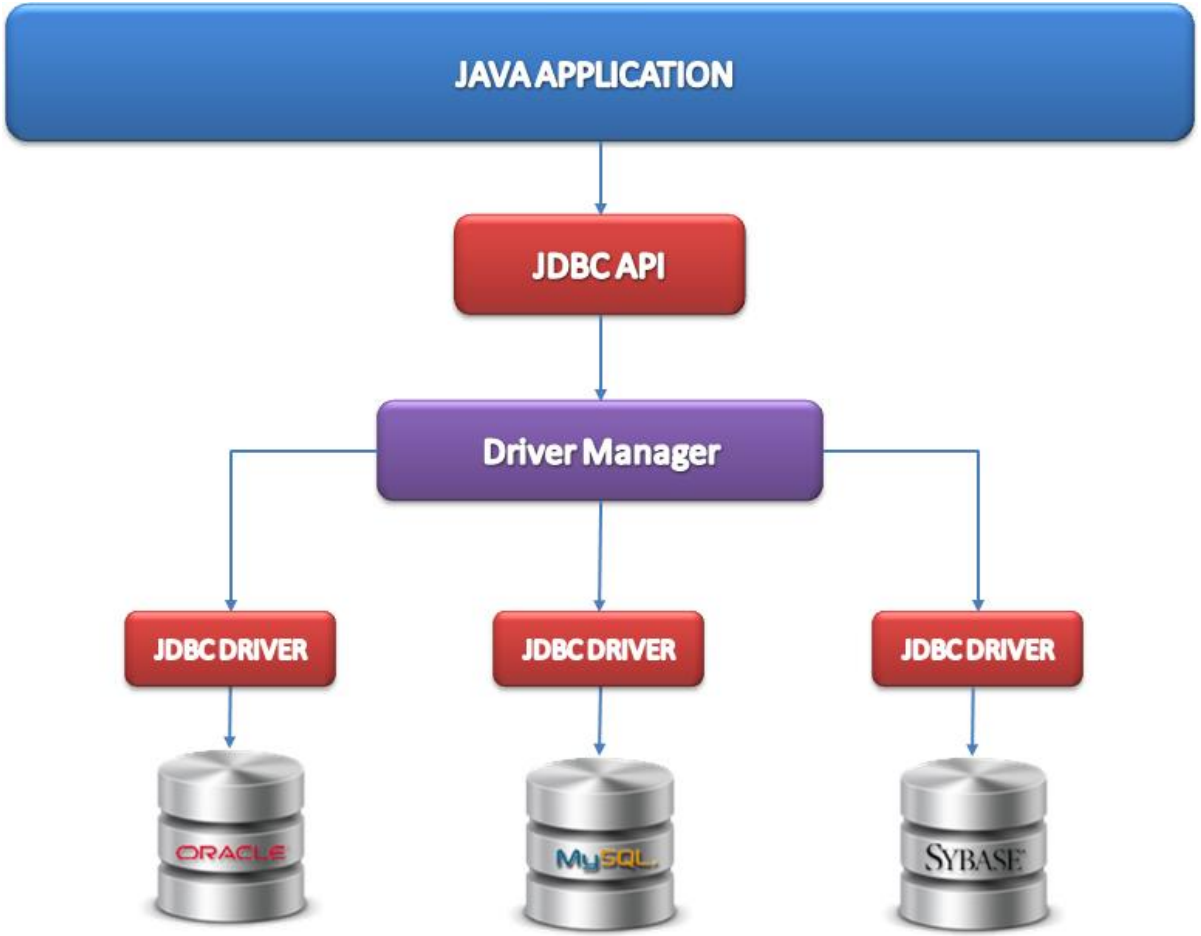
JAVA 1

JDBC, XML

Teme

- JDBC
- Driver & Connection
- DataSource
- Statement & ResultSet
- XML Parsers
- Sax, Stax, Dom
- JaxB

JDBC



Izvor:https://avaldes.com/connecting-to-sqlserver-using-jdbc/

JDBC

- **J**ava **D**ata**b**ase **C**onnectivity
- standardni Java API za konekciju prema bazi podataka
- neovisan o bazi podataka – most je *driver*
- pomoću *driver*-a komunicira s bazom podataka korištenjem upita (engl. *query*)
- *driver* upite pretvara u određeni komunikacijski protokol svojstven bazi podataka
- potrebno referencirati *driver* kroz Maven dependencies

JDBC

Izvor:https://www.mysqltutorial.org/jdbc-overview/

Driver & connection

```
Connection con = null;
try {
    Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
    String conUrl = "jdbc:sqlserver://localhost/mydatabase";
    con = DriverManager.getConnection(conUrl, "root", "password");
} catch (ClassNotFoundException | SQLException ex) {
} finally {
    try {
        if (con != null) {
            con.close();
        }
    } catch (SQLException ex) {
    }
}
```

Driver & Connection

- učitavanje *driver*-a pomoću refleksije
- *driver* se u svom statičkom bloku registrira za *DriverManager*
- dohvaćanje konekcije preko *DriverManager*-a, pružajući mu *connectionString*
- konekcija se nakon korištenja mora zatvoriti
- presloženo i nepotrebno – koristimo *DataSource*

DataSource

```
public final class DataSourceSingleton {
    private static final String SERVER_NAME = "localhost";
    private static final String DATABASE_NAME = "Vjezbe0708";
    private static final String USER = "sa";
    private static final String PASSWORD = "SQL";

    private DataSourceSingleton() {}

    private static DataSource instance;

    public static DataSource getInstance() {
        if (instance == null)  instance = createInstance();
        return instance;
    }
    private static DataSource createInstance() {
        SQLServerDataSource dataSource = new SQLServerDataSource();
        dataSource.setServerName(SERVER_NAME);
        dataSource.setDatabaseName(DATABASE_NAME);
        dataSource.setUser(USER);
        dataSource.setPassword(PASSWORD);
        return dataSource;
    }
}
```

DataSource

```
DataSource dataSource = DataSourceSingleton.getInstance();
try (Connection con = dataSource.getConnection()) {
} catch (Exception e) {
}
```

Statement & ResultSet

- **Statement**
- služi za izvršavanje upita
- *PreparedStatement* – parametrizirani upiti
- *CallableStatement* – parametrizirani pozivi procedura i funkcija
- **ResultSet**
- rezultat upita
- sadrži retke koji zadovoljavaju upit

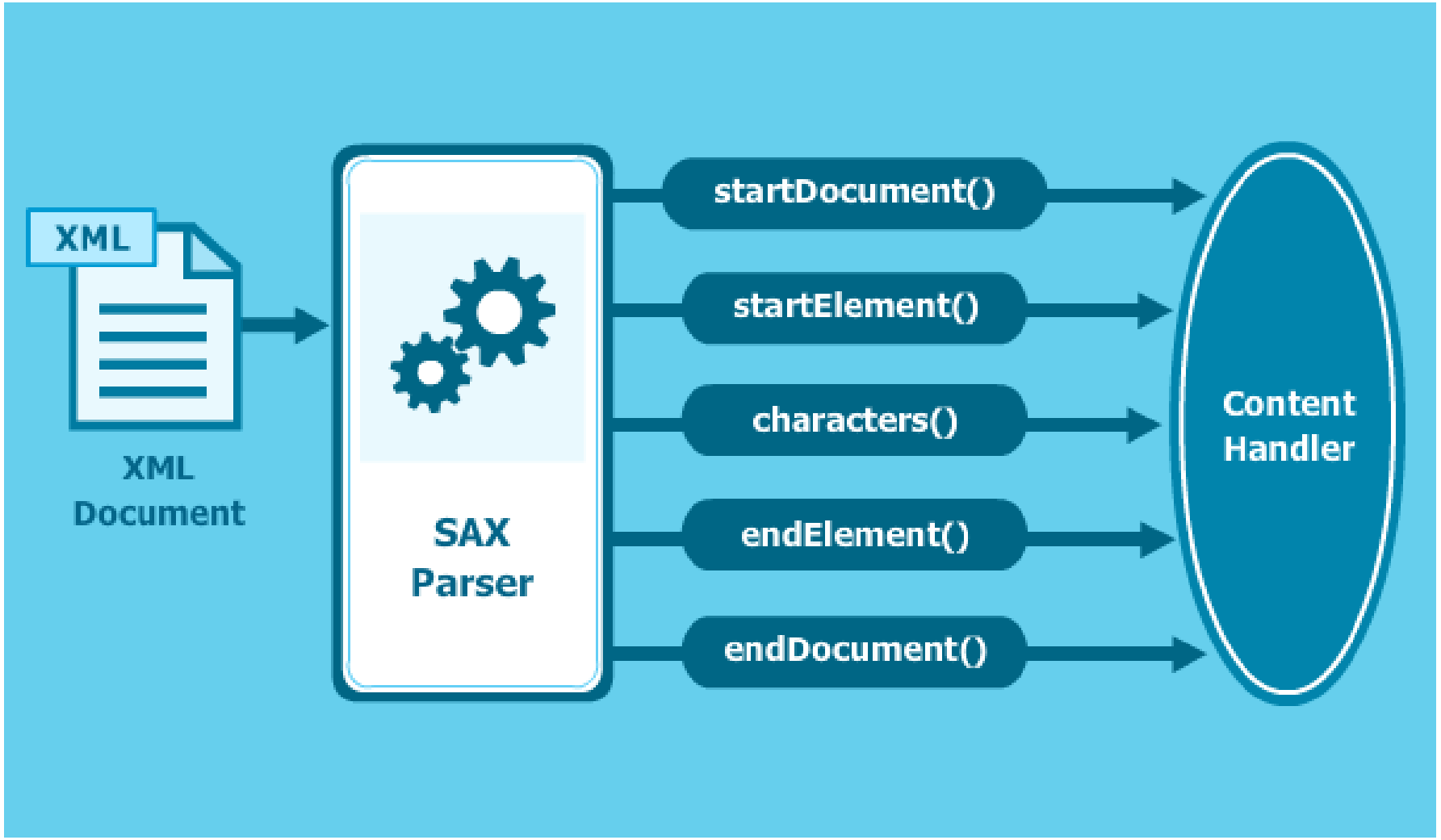
Statement & ResultSet

```
@Override
public List<Article> selectArticles() throws Exception {
    List<Article> articles = new ArrayList<>();
    DataSource dataSource = DataSourceSingleton.getInstance();
    try (Connection con = dataSource.getConnection();
            CallableStatement stmt = con.prepareCall(SELECT_ARTICLES);
            ResultSet resultSet = stmt.executeQuery()) {
        while (resultSet.next()) {
            articles.add(
                    new Article(
                            resultSet.getInt("IDArticle"),
                            resultSet.getString("Title"),
                            resultSet.getString("Link"),
                            resultSet.getString("Description"),
                            resultSet.getString("PicturePath"),
                            LocalDateTime.parse(resultSet.getString("PublishedDate"), Article.DATE_FORMATTER)));
        }
    } catch (Exception e) {
        Logger.getLogger(SqlRepository.class.getName()).log(Level.SEVERE, null, e);
    }
    return articles;
}
```

XML Parsers

- **Sax** - **S**imple **A**PI for **X**ML
- *push model* - parsira liniju po liniju i obaviještava pretplatnike o događajima (otvaranje, zatvaranje tag-a, čitanje teksta...)
- **Stax** - **S**treaming **A**PI for **X**ML
- *pull model* – klijent sam *povlači* događaje
- **Dom** - **D**ocument **O**bject **M**odel
- učitava cijeli xml u memoriju i omogućuje prolazak po čvorovima

Sax



Izvor:https://www.edureka.co/blog/parsing-xml-file-using-sax-parser/

Stax

Izvor:https://www.developerfusion.com/article/84523/stax-the-odds-with-woodstox/

Dom

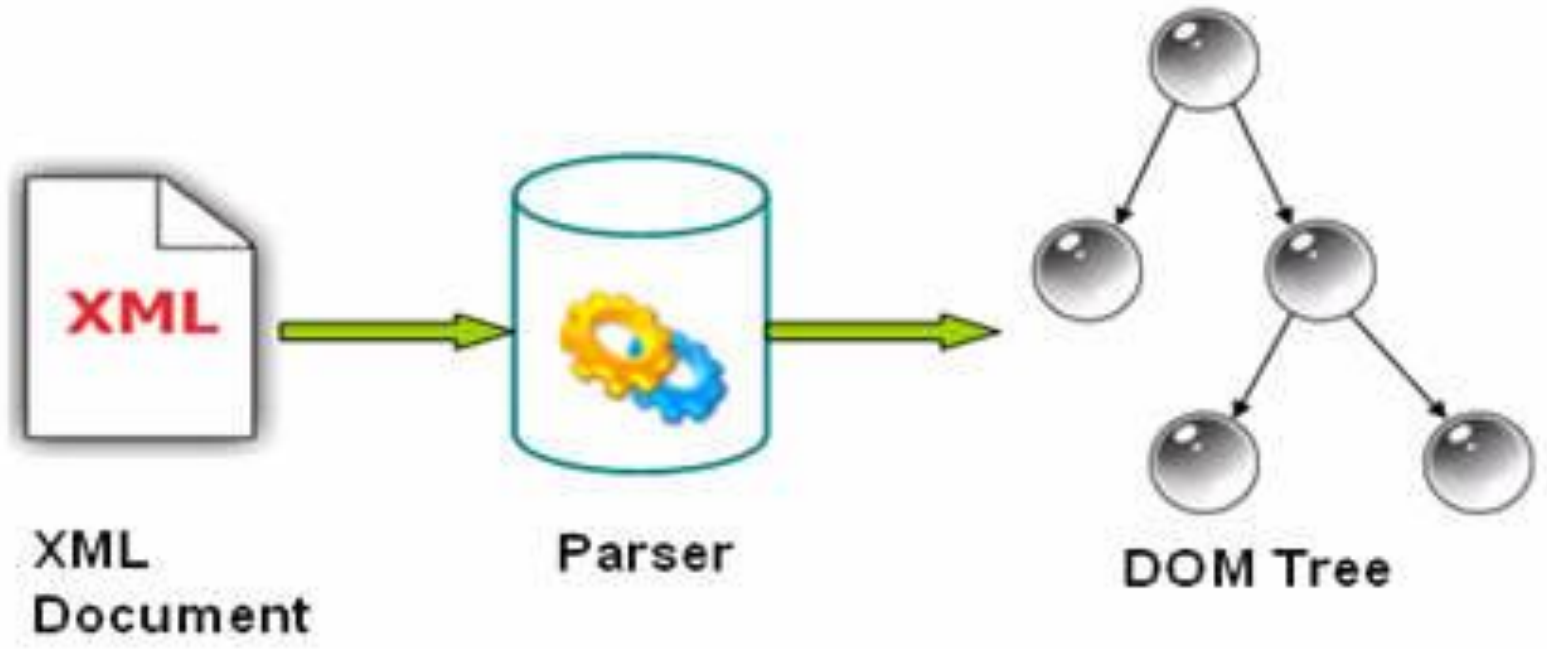


Fig a. Document Object Model

Izvor:https://www.topjavatutorial.com/java/java-programs/parsing-xml-dom-java/

JaxB

- **J**ava **A**rchitecture for **X**ML **B**inding
- *bind*-a, povezuje Java kod i xml tagove
- anotacije – mapiranje klasa sa njihovim xml reprezentacijama
- lagana prilagodba – imena, atributi, liste…
- *marshall* – omogućuje zapisivanje objekata u xml
- *unmarshall* – obrnut postupak za kreiranje objekata iz xml

Demo

- Project

Izvor:http://www.jnhsolutions.com/contact-us/request-a-demo/

Hvala na pažnji!