

2. Use NETCONF4J for NETCONF operations like edit-config, get, and delete to configure and retrieve device configurations

Include **NETCONF4J** in your project dependencies.

```
<dependency>
  <groupId>io.frxn.cli</groupId>
  <artifactId>netconf4j</artifactId>
  <version>3.0.0</version>
</dependency>
```

Basic NETCONF Operations like `get`, `edit-config`, `delete-config` can be done as follows:

```
import io.frxn.cli.io.Cli;
import io.frxn.cli.io.impl.NetconfSession;
import io.frxn.cli.io.impl.NetconfSessionBuilder;
import org.opendaylight.yang.gen.v1.urn.ietf.params.xml.ns.yang.ietf.netconf.rev110601.Get;
import org.opendaylight.yang.gen.v1.urn.ietf.params.xml.ns.yang.ietf.netconf.rev110601.EditConfig;
import org.opendaylight.yang.gen.v1.urn.ietf.params.xml.ns.yang.ietf.netconf.rev110601.DeleteConfig;

public class NETCONFManager {

    private NetconfSession netconfSession;

    public NETCONFManager(String host, int port, String username, String password) throws Exception
    {
        netconfSession = new NetconfSessionBuilder()
            .setHost(host)
            .setPort(port)
            .setUsername(username)
            .setPassword(password)
            .build();
    }

    // Perform NETCONF get operation
    public String getConfiguration(String filterXml) throws Exception {
        Get getRequest = new Get();
        getRequest.setFilter(filterXml);
        return netconfSession.get(getRequest);
    }

    // Perform NETCONF edit-config operation
    public void editConfiguration(String target, String configXml) throws Exception {
        EditConfig editConfigRequest = new EditConfig();
        editConfigRequest.setTarget(target);
        editConfigRequest.setConfig(configXml);
        netconfSession.editConfig(editConfigRequest);
        System.out.println("Configuration edited successfully.");
    }

    // Perform NETCONF delete-config operation
    public void deleteConfiguration(String target) throws Exception {
        DeleteConfig deleteConfigRequest = new DeleteConfig();
        deleteConfigRequest.setTarget(target);
        netconfSession.deleteConfig(deleteConfigRequest);
    }
}
```

```

        System.out.println("Configuration deleted successfully.");
    }

    public static void main(String[] args) {
        try {
            NETCONFManager manager = new NETCONFManager("192.168.1.1", 830, "username", "password");

            // Get device configuration
            String filterXml = "<interfaces xmlns='urn:ietf:params:xml:ns:yang:ietf-interfaces'/>";
            String response = manager.getConfiguration(filterXml);
            System.out.println("Configuration retrieved: " + response);

            // Edit device configuration
            String editConfigXml = ""
                <interfaces xmlns='urn:ietf:params:xml:ns:yang:ietf-interfaces'>
                    <interface>
                        <name>eth0</name>
                        <description>Configured by NETCONF</description>
                        <enabled>true</enabled>
                    </interface>
                </interfaces>
            "";
            manager.editConfiguration("running", editConfigXml);

            // Delete device configuration
            manager.deleteConfiguration("candidate");

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```