README

gRPC in Springboot Java

Author: Rahul Gupta
Date: Feb, 14

Explanation of the Project

Getting started

In order to utilize gRPC with Springboot we need to use gradle and define the necassary code blocks inside the build.gradle file. With spring initializr0

Generating Java files out of .proto file

`./gradlew generateProto`

Common Issues and Solutions

1. **Importing the Generated Class Issue**: The generated `PersonOuterClass` couldn't be imported. This was resolved by adjusting the `sourceSets` in `build.gradle` to include the directory of the generated sources, so the IDE could recognize them as part of the project. In the `build.gradle` file, it was:

```
sourceSets {
    main {
       java {
          srcDirs 'build/generated/source/proto/main/java', 'src/main/java'
       }
    }
}
```

With this, `PersonOuterClass` could be imported with: `import com.example.proto.PersonOuterClass;`

2. **`generateProto` Execution Error**: Attempting to run the `generateProto` task failed due to an unresolved dependency. This was caused by using a release candidate version (`4.26.0-RC2`) of `protobuf-java`, which might have been incompatible or not available.

The solution was to switch to a stable version, which was **3.25.2** in this case, and restarting the IDE resolved the issue.

3. **Missing Package Declaration**: Initially, the generated class couldn't be imported because the `.proto` file lacked a package declaration, causing the generated class to default to the default package. The issue was fixed by adding a `package` statement to the `.proto` file and re-running the `generateProto` task to regenerate the classes with the correct package name. For this case, it was **package com.example.proto;**

```
syntax = "proto3";

package com.example.proto;

message Person {
  optional string name = 1;
  optional int64 id = 2;
  optional string email = 3;
}
```