

## 02. gRPC 실용사례 Node.js

본 문서에서는 node.js 에서 gRPC 를 사용하여 Client 프로그램을 작성 하는 방법을 소개 한다. 사전 조건 으로 node.js 및 npm 이 설치 되어 있어야 한다.

### 1. Interface 를 위한 .proto 파일

실용사례 에서 공통으로 사용할 .proto 파일로 이름은 `xdb_grpc.proto` 로 작성 하였다.

01. gRPC 실용사례 CPP 에 있는 내용과 동일 하다.

```
syntax = "proto3";

// Option for Java.
option java_multiple_files = true;
option java_package = "com.hancomsechre.xecuredb.grpc";
option java_outer_classname = "XecuredbProto";

// Option for Objective-C.
option objc_class_prefix = "XDB";

package xdbproto;

// The Xecuredb service definition.
service Xecuredb {
  rpc Encrypt (EncRequest) returns (EncReply) {}
  rpc Decrypt (DecRequest) returns (DecReply) {}
  rpc CryptoHash (HashRequest) returns (HashReply) {}
}

// The request message containing the alias and plain text.
message EncRequest {
  string alias = 1;
  string plain = 2;
}

// The response message containing the err code and cipher text.
message EncReply {
  int32 err = 1;
  string msg = 2;
  string cipher = 3;
}

// The request message containing the alias and cipher text.
message DecRequest {
  string alias = 1;
  string cipher = 2;
}

// The response message containing the err code and plain text.
message DecReply {
  int32 err = 1;
  string msg = 2;
}
```

```

    string plain = 3;
}

// The request message containing the alogorithm id and plain text.
message HashRequest {
    int32 id = 1;
    string plain = 2;
}

// The response message containing the err code and hash string.
message HashReply {
    int32 err = 1;
    string msg = 2;
    string cipher = 3;
}

```

## 2. 종속성 Package download

node.js 에서 gRPC 를 사용하기 위해서는 종속성 Package 를 받아와야 한다. package 를 받아오기 위해 서는 npm(node package manager) 를 이용 하여야 하는데 npm install 명령은 기본적으로 현재 directory 의 `package.json` 에 기록 되어 있는 파일을 참조하여 package 를 설치 한다. 아래 에는 Client 를 작성 하기 위한 `package.json` 이다.

```

{
  "name": "node_xdb_client",
  "version": "0.1.0",
  "dependencies": {
    "@grpc/proto-loader": "^0.1.0",
    "async": "^1.5.2",
    "google-protobuf": "^3.0.0",
    "grpc": "^1.11.0",
    "lodash": "^4.6.1",
    "minimist": "^1.2.0"
  }
}

```

npm 을 이용하여 종속성 package 설치 명령은 다음과 같다.

```
npm install
```

위 명령을 수행 하면 현재 경로 아래 `node_modules/` 경로가 생성 되며 그 아래 연관된 package 들이 설 치 된다. 그리고 `package-lock.json` 파일도 추가로 생성 된다.

## 3. Client 코드 작성

node.js 를 이용한 client 는 `node_xdb_client.js` 로 작성 하였으며 source code 는 다음과 같다.

```

var PROTO_PATH = __dirname + '/../proto/xdb_grpc.proto';

var grpc = require('grpc');
var protoLoader = require('@grpc/proto-loader');

```

```

var packageDefinition = protoLoader.loadSync(
  PROTO_PATH,
  {keepCase: true,
    longs: String,
    enums: String,
    defaults: true,
    oneofs: true
  });
var xdb = grpc.loadPackageDefinition(packageDefinition).xdbproto;

function main() {
  var client = new xdb.Xecuredb('localhost:50052',
                                grpc.credentials.createInsecure());

  var in_alias = 'normal';
  var in_plain = '1234567890123';
  var in_cipher = '';

  // 암호화 함수 호출
  client.Encrypt({alias: in_alias, plain: in_plain}, function(err, response) {
    console.log('Encrypt:', response.cipher);
    in_cipher = response.cipher;

    // 복호화 함수 호출
    client.Decrypt({alias: in_alias, cipher: in_cipher}, function(err, response)
  {
    console.log('Decrypt:', response.plain);
  });
});

// Hash 함수 호출
client.CryptoHash({id: 6, plain: in_plain}, function(err, response) {
  console.log('Hash:', response.cipher);
});

}

main();

```

Node.js 에서 gRPC 호출은 별도의 Interface 모듈을 protoc 를 통하여 생성 하지 않고 .proto 파일을 protoLoader.loadSync() 함수를 통해 load 한 후 사용하는 간단한 방법이 있다.

위의 예제를 보면 `var PROTO_PATH = __dirname + '/../proto/xdb_grpc.proto';` 에서 직접 .proto 파일을 지정 하여 사용 하는것을 볼 수 있다. 그리고 `grpc.loadPackageDefinition()` 를 통해 package 를 load 하고 client 를 생성 한 후 바로 사용 하는 모습을 볼 수 있다. 약간의 주의만 기울이면 쉽게 gRPC Client 를 만들 수 있다. Node.js 의 특성상 함수가 호출 한 순서 대로 실행 되지 않는다. 내 Vmware 에 설치된 Ubuntu 에서는 Hash, Encrypt, Decrypt 순으로 표시 된다.

## 4. node.js client 프로그램 실행

프로그램의 실행은 다음 명령을 이용한다.

```
node node_xdb_client.js
```

#### 실행 결과

```
Hash: vKK0Gis14TfIP+40ave9Hg9SvVYFg8oHobQvmUTFxQs=  
Encrypt: AAGo8vet8aHoqsGorhx1CsXL  
Decrypt: 1234567890123
```

## 5. Remark

위 프로그램은 gRPC examples 에 있는 Helloworld 를 참조 하여 작성 하였다. protoc 를 이용한 Interface 코드는 생성 해 주지 않지만 지원되는 package 가 있어 타 언어 보다 쉽게 구현이 가능 하다.

- 끝 -