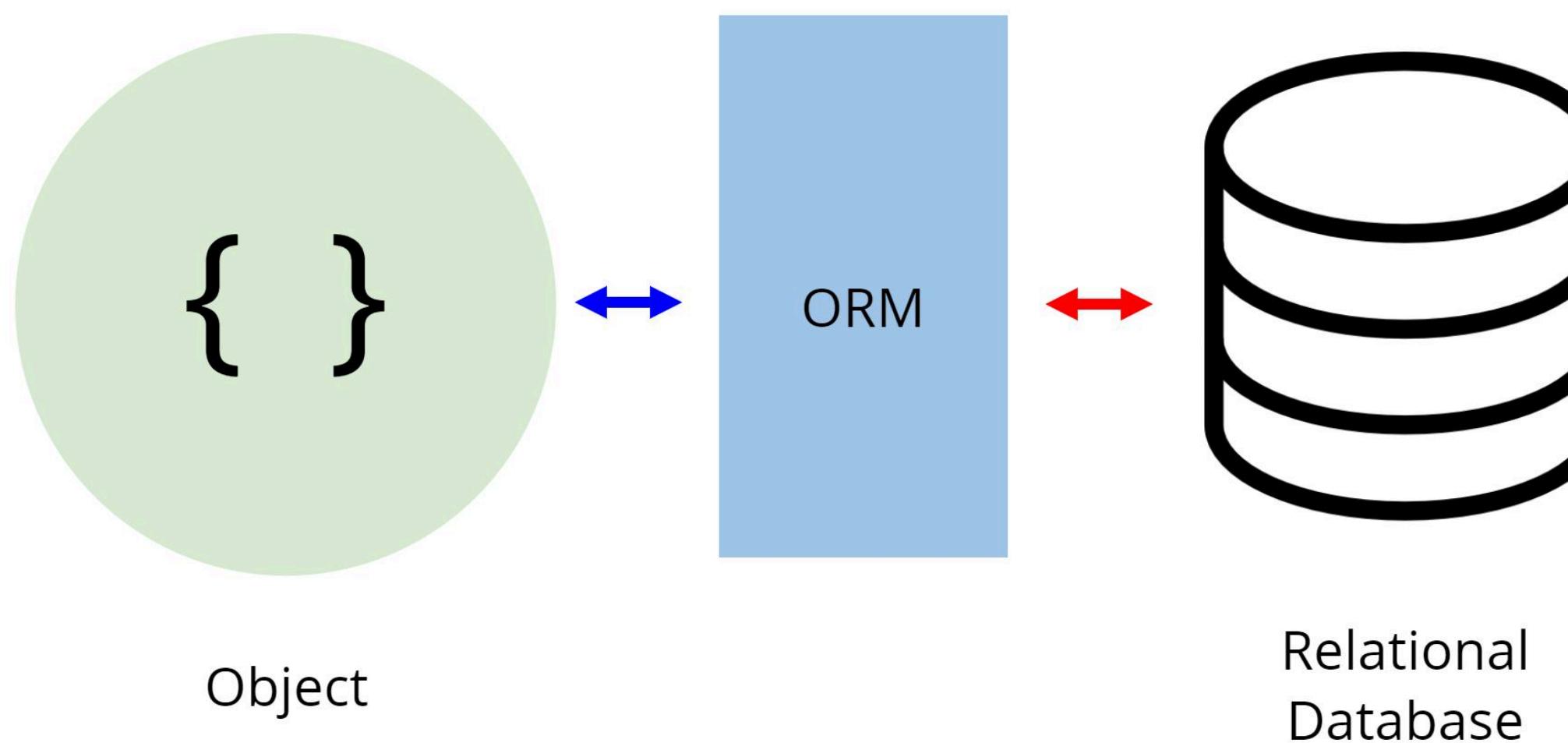


[5주차] Node JS to TS

이윤슬



ORM이란?



Object Relational Mapping

- 객체 지향 프로그래밍 언어와 관계형 데이터베이스의 호환되지 않는 유형 시스템 간에 데이터를 변환하는 데 사용되는 프로그래밍 기술
- 비유하면, 객체지향 세계와 RDB 세계를 이어주는 다리!

ORM의 장점



```
// SQL 없이 JS로 데이터 조작 가능  
const user = await User.create({ name: 'Yunseul' });
```

**SQL 없이 JS 코드로 DB 조작
유지보수성 향상, 재사용 용이**

모델이란?

- ORM에서 모델은 테이블의 객체 표현
- JS 클래스 ↔ DB 테이블 간 추상화

방법		
<code>sequelize.define()</code>	간결, 초기 예제에 적합	함수형 선언
<code>class extends Model</code>	확장성 높음	객체지향적 구조

모델이란?

sequelize.define() 방식 → 함수형 선언
(공식문서에서 첫 번째 방법)

Enforcing the table name to be equal to the model name

You can stop the auto-pluralization performed by Sequelize using the `freezeTableName: true` option. This way, Sequelize will infer the table name to be equal to the model name, without any modifications:

```
sequelize.define(
  'User',
  {
    // ... (attributes)
  },
  {
    freezeTableName: true,
  },
);
```

The example above will create a model named `User` pointing to a table also named `User`.

This behavior can also be defined globally for the sequelize instance, when it is created:

```
module.exports = (sequelize, DataTypes) => {
  const Review = sequelize.define('Review', {
    title: {
      type: DataTypes.STRING(100),
      allowNull: false,
    },
    content: {
      type: DataTypes.TEXT,
      allowNull: false,
    }
  }, {
    tableName: 'reviews',
    timestamps: true
  });

  return Review;
};
```

Sequelize 초기화 및 자동 로딩



```
const basename = path.basename(__filename);
const config = require(__dirname + '/../config/config.json')
[env];
const db = {};

let sequelize;
if (config.use_env_variable) {
  sequelize = new
Sequelize(process.env[config.use_env_variable], config);
} else {
  sequelize = new Sequelize(config.database, config.username,
config.password, config);
}
```



```
fs.readdirSync(__dirname)
.filter(file => file !== basename && file.endsWith('.js'))
.forEach(file => {
  const model = require(path.join(__dirname, file))(sequelize,
Sequelize.DataTypes);
  db[model.name] = model;
});
```

Express 라우터 구조



```
router.get('/', (req, res) => {
  const reviews = reviewService.getAllReviews();
  res.json(reviews);
});
```



```
const { Review } = require('../models');

exports.getAllReviews = async () => {
  return await Review.findAll({ order: [['createdAt', 'DESC']] });
};
```

Raw Query vs ORM



```
await sequelize.query('SELECT * FROM reviews'); // Raw SQL  
await Review.findAll(); // ORM
```

QnA

자유롭게 질문해주세요.



Node.js

조현영 지음