작업 일지 22.11.21

> ==login 관련 simple spring boot restApi server 만들기.==

DB connection 구현하기

> == 상세 ==
>
> 1. db connection 정보 만들기
>
> - rest > src > main > resources > application.properties

```
# db connection 정보
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:3306/carutil?characterEncoding=UTF-8
spring.datasource.username=naru
spring.datasource.password=3031
spring.datasource.tomcat.max-active=30
spring.datasource.tomcat.max-idle=20
spring.datasource.tomcat.min-idle=10
```

> 2. db config.java 만들기 (connection 정보 mapper 들과 일치 시키기)
>
> - rest > src > main > java > com > carutil > rest > config > DbConfig.java
> - 여기서의 mapper 는 xml 이 아닌 interface 가 있는 mapper.

```
// DbConfig.java
package com.carutil.rest.config;

import javax.sql.DataSource;

import org.apache.ibatis.session.SqlSessionFactory;
import org.mybatis.spring.SqlSessionFactoryBean;
import org.mybatis.spring.SqlSessionTemplate;
import org.mybatis.spring.annotation.MapperScan;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.core.io.support.PathMatchingResourcePatternResolver;

// 자동 실행 하기 위해 @Configuration 설정
@Configuration

// mapper 가 있는 위치 지정 ( interface 로서의 mapper 위치 )
@MapperScan(basePackages = "com.carutil.rest.mapper")
public class DbConfig {

    // bean Object 생성을 위해 필요한 annotation
    @Bean
    public SqlSessionFactory sqlSessionFactory(DataSource dataSource) throws
```

```
Exception {

        SqlSessionFactoryBean sqlSessionFactory = new SqlSessionFactoryBean();

        sqlSessionFactory.setDataSource(dataSource);

        PathMatchingResourcePatternResolver resolver = new
PathMatchingResourcePatternResolver();


sqlSessionFactory.setMapperLocations(resolver.getResources("classpath:mapper/**/*.
xml"));

        return sqlSessionFactory.getObject();

    }

    @Bean
    public SqlSessionTemplate sqlSessionTemplate(SqlSessionFactory
sqlSessionFactory) throws Exception {
        return new SqlSessionTemplate(sqlSessionFactory);
    }

}
```

3. mapper interface 만들기 (memeber)

- rest > src > main > java > com > carutil > rest > mapper > MemberMapper.java
- 그외 다수의 table 별 interface 들을 필요에 따라 생성

```
// MemberMapper.java
// Member interface simple example
package com.carutil.rest.mapper;

import org.apache.ibatis.annotations.Param;

import com.carutil.rest.vo.MemberVO;

public interface MemberMapper {

    // 현재 interface 들을 연결하고 있는 sql mapper xml 에 정의 되어있는
    // 각 query 들의 id 를 abstract method 로 정의 한다.

    MemberVO login(@Param("id") String id,
            @Param("pw") String pw);
}
```

4. service 만들기

- rest > src > main > java > com > carutil > rest > service > MemberService.java

> - 그외 다수의 table 별 interface 들을 필요에 따라 생성

```
package com.carutil.rest.service;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.carutil.rest.mapper.MemberMapper;
import com.carutil.rest.vo.MemberVO;

// @Service 라고 명시해야 bean 객체로 인식하고,
// MemberMapper interface 를 인식한다.
@Service
public class MemberService {

    @Autowired
    private MemberMapper mapper;

    public MemberVO login(String id, String pw) {

        return mapper.login(id, pw);
    }
}
```

 # 회원 관련 구현하기

> ==상세==
>
> 1. spring boot 에서 회원 관련 db table 과 일치 시킬 VO 준비
>
> - rest > src > main > java > com > carutil > rest > vo > MemberVO.java
>
> 2. mapper (mybatis xml) 준비
>
> - rest > src > main > resources > mapper > member.xml

```
// MemberVO.java
package com.carutil.rest.vo;

import lombok.AllArgsConstructor;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;

@Setter
@Getter
@NoArgsConstructor
@AllArgsConstructor
public class MemberVO {
    private String m_idx;
```

```
    private String id;
    private String pw;
    private String email;
    private String name;
    private String nick;
    private String birth;
    private String phone;
    private String snsAuth;
    private String snsID;
    private String manner_point;
    private String activate;
    private String grade;
    private String regDate;
}
```