



Spring_RESTful API

```
@RestController
public class carController {

    private modelMapper mapper;

    public carController(modelMapper mapper) {
        this.mapper = mapper;
    }

    // FOR CHART
    @GetMapping("/carmodel/{cseq}")
    public carModel getCarModelInformation(@PathVariable("cseq")String id) {
        return mapper.getCarModelInformation(id);
    }

    // FOR CHART LIST USE THIS
    @GetMapping("/carmodel/list/{model}")
    public List<carModel> getCarAudiModelList(@PathVariable("model")String model){
        return mapper.getCarAudiModelList(model);
    }

    // FOR CHART LIST USE THIS
    @GetMapping("/carmodel/VWlist/{model}")
    public List<carModel2> getCarVWModelList(@PathVariable("model")String model){
        return mapper.getCarVWModelList(model);
    }
}
```

Spring Rest api를 통해 flutter 에서 작동하는 Query문을 스프링에서 처리한다.

```
@Mapper
public interface modelMapper {

    // 특정 cseq 값 내용 선택
    @Select("SELECT * FROM audi WHERE cseq =#{cseq}")
    carModel getCarModelInformation(@Param("cseq")String id);

    @Select("SELECT * FROM audi WHERE model=#{model}")
    List<carModel> getCarAudiModelList(@Param("model")String model);

    @Select("SELECT * FROM vw_tiguan WHERE model=#{model}")
    List<carModel2> getCarVWModelList(@Param("model")String model);
}
```

Controller를 통해 입력받은 값을 Mapper에서 DB에 연결하여 원하는 Query문을 실행한 후. Flutter를 View 페이지로 사용하여 결과를 보여준다.



Spring_RESTful API

```
@RestController
public class rdsController {

    @RequestMapping(value = "/urlcar", produces = "application/json; charset=utf8", method = RequestMethod.GET)
    public JSONObject Abouttp(HttpServletRequest request) throws Exception{
        request.setCharacterEncoding("utf-8");
        double year = Double.parseDouble(request.getParameter("year"));
        double mileage = Double.parseDouble(request.getParameter("mileage"));
        double engineSize = Double.parseDouble(request.getParameter("engineSize"));
        double mpg = Double.parseDouble(request.getParameter("mpg"));
        String Manual = request.getParameter("Manual");
        String fuelType_D = request.getParameter("fuelType_D");
        String fuelType_p = request.getParameter("fuelType_p");
        String fileName = request.getParameter("fileName");
        RConnection conn1 = new RConnection();

        conn1_voidEval("library(randomForest)");
        conn1_voidEval("library(e1071)");
        String rdsRoot = System.getProperty("user.dir", "UTF-8") + "/src/main/resources/webapp/rds/";

        conn1_voidEval("rfs <- readRDS(\"\" + rdsRoot + fileName + "\", 'rb')");
        conn1_voidEval("result <- as.character(predict(rfs, (list(year=" + year + ", mileage=" + mileage + ", "
+ "engineSize=" + engineSize + ", mpg=" + mpg + ", Manual=" + Manual +
", fuelType_D=" + fuelType_D + ", fuelType_P=" + fuelType_p + "))))");

        String result2 = conn1.eval("result").asString();

        Map<String, Object> map = new HashMap<>();
        map.put("result", result2);

        JSONObject jsonObject2 = new JSONObject(map);
        conn1.close();
        return jsonObject2;
    }
}
```

Rserve Api를 이용해 스프링과 R을 연결하여 Flutter에서 값을 입력할 시 연결된 Rds 파일을 통해 예측 값을 리턴해 준다.