◎ 인터페이스 활용예제

```
public interface GoogleApi {
	public double[] 위도경도구하기();
}
```

```
@Component
public class GoogleApiTemp  implements GoogleApi {

	@Override
	public double[] 위도경도구하기() {
		double[]  result = { 37.4997187    ,127.0348982};
		return result;

	}

}
```

```java
@Component
public class GoogleApiImp  implements GoogleApi {

	@Override
	public double[] 위도경도구하기() {
			String address = "서울특별시 강남구 역삼동 736-1";
			// Google Maps Platform에서 발급받은 API 키를 입력합니다.
			String apiKeys = "AIzaSyAsLpi_vMOsUZeIv2OpKyosmuE1i-kOOrY";

			// Geocoding API의 URL을 생성합니다.
			String apiUrl="";
				try {
					apiUrl = "https://maps.googleapis.com/maps/api/geocode/json" +
							"?address=" + URLEncoder.encode(address, "UTF-8") +
							"&key=" + apiKeys;
				} catch (UnsupportedEncodingException e) {
					// TODO Auto-generated catch block
					e.printStackTrace();
				}

			// API를 호출하여 위도와 경도 정보를 가져옵니다.
			URL url;
			StringBuilder response=null;
				try {
					url = new URL(apiUrl);
				HttpURLConnection connection = (HttpURLConnection) url.openConnection();
					connection.setRequestMethod("GET");

					BufferedReader reader = new BufferedReader(new
InputStreamReader(connection.getInputStream()));
				response = new StringBuilder();
				String line;

					while ((line = reader.readLine()) != null) {
					    response.append(line);
					}
					reader.close();

				} catch (MalformedURLException e) {
					// TODO Auto-generated catch block
					e.printStackTrace();
				} catch (ProtocolException e1) {
					// TODO Auto-generated catch block
					e1.printStackTrace();
				} catch (IOException e1) {
					// TODO Auto-generated catch block
					e1.printStackTrace();
				}


				System.out.println( response.toString() );


			// 위도와 경도 정보를 출력합니다.
			JSONObject jsonResponse = new JSONObject(response.toString());
			JSONObject location = jsonResponse.getJSONArray("results")
								.getJSONObject(0)
								.getJSONObject("geometry")
								.getJSONObject("location");
			double latitude = location.getDouble("lat");
			double longitude = location.getDouble("lng");
			System.out.println("위도: " + latitude + ", 경도: " + longitude);

			double result[]    = new double[2];
			result[0]= latitude;
			result[1]= longitude;

				System.out.println( latitude+ "   " +  longitude );


			  return result;

	}

}
```

```java
public interface TestServiceI {

	public  double[]  위도경도가지오기() ;
	public  int registerMember();

}


@Component
public class TestServiceImp implements TestServiceI {
	/*
	public static void main(String[] args) {
		TestServiceImp  s = new TestServiceImp();
		double[] result = s.위도경도가지오기();

	}
	*/

	@Autowired
	 GoogleApi  api;

	@Autowired
	TestRepository  repository;


	@Override
	public double[] 위도경도가지오기() {
		double[] result  = api.위도경도구하기();
		return result;
	}


	@Override
	public int registerMember() {
		double[] result  = 위도경도가지오기();
		System.out.println( result[0] + " : " + result[1]);
		Member m =new Member();
		repository.insertMember(m);
		return 1;
	}

}
```

```java
public class Member {

	String id;
	String name;
	double latitude ;
	double longitude;

	public String getId() {
		return id;
	}
	public void setId(String id) {
		this.id = id;
	}
	public String getName() {
		return name;
	}
	public void setName(String name) {
		this.name = name;
	}
	public double getLatitude() {
		return latitude;
	}
	public void setLatitude(double latitude) {
		this.latitude = latitude;
	}
	public double getLongitude() {
		return longitude;
	}
	public void setLongitude(double longitude) {
		this.longitude = longitude;
	}
	@Override
	public String toString() {
		return "Member [id=" + id + ", name=" + name + ", latitude=" + latitude + ", longitude=" + longitude +
"]";
	}
	public Member(String id, String name, double latitude, double longitude) {
		super();
		this.id = id;
		this.name = name;
		this.latitude = latitude;
		this.longitude = longitude;
	}


    public Member() {
		// TODO Auto-generated constructor stub
	}

}
```

```java
@Controller
public class TestController {

	@Autowired
	TestServiceImp service;


	@RequestMapping( value="/member")
	public String test( ) {
		return  "register";
	}



	@RequestMapping( value="/member" , method =RequestMethod.POST )
	public void test(Member member) {
		service.registerMember( member);
	}

}
```

```java
@Component
public class TestRepository {

	@Autowired
	DataSource ds;


	public  int insertMember(  Member member) {

		//sql작성  데이터베이스에 insert 하기
		int resultRow=0;
		try {
			String sql="insert into  membertest2  values(  ?,?,?,?) ";
			Connection con  = ds.getConnection();
		PreparedStatement pst  =   con.prepareStatement(sql)        ;
		pst.setString(1, member.getId());
		pst.setString(2, member.getName());
		pst.setDouble(3,  member.getLatitude());
		pst.setDouble(4,   member.getLongitude());
		resultRow  = pst.executeUpdate();
		close(pst, con);
		return resultRow;
		} catch (SQLException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
	return  resultRow;
	}


	private void close( AutoCloseable ...autoCloseables ) {
		for( AutoCloseable obj : autoCloseables) {
			try {
				obj.close();
			} catch (Exception e) {
				// TODO Auto-generated catch block
				e.printStackTrace();
			}
		}
	}

}
```

```
package com.acorn.test2;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;

@Controller
public class MyController {

	@Autowired
	MyService service;


	@GetMapping("/memberT")
	public String  register() {
		return "memberT";
	}


	@PostMapping("/memberT")
	public String  register(Member member) {
		service.registerMember(member);
		return "home";
	}

}
```

```java
public interface MyService {

	public int  registerMember( Member member);
	public List<Member>  getMemberList();
	public int  modifyMember( Member member);
	public int  deleteMember(String userId);

}

public interface MyRepository {

	public int  insert( Member member);
	public List<Member>  selectAll();
	public int  update( Member member);
	public int  delete(String userId);



}


@Service    //@Component
public class ServiceImp  implements MyService{

	@Autowired
	MyRepository dao;

	@Override
	public int registerMember(Member member) {
		// TODO Auto-generated method stub
		return dao.insert(member);
	}

	@Override
	public List<Member> getMemberList() {
		// TODO Auto-generated method stub
		return dao.selectAll();
	}

	@Override
	public int modifyMember(Member member) {
		// TODO Auto-generated method stub
		return  dao.update(member);
	}

	@Override
	public int deleteMember(String userId) {
		// TODO Auto-generated method stub
		return dao.delete(userId);
	}

}



@Repository    //@Component
public class RepositoryImp  implements MyRepository {

	@Override
	public int insert(Member member) {
		// TODO Auto-generated method stub
		return 0;
	}

	@Override
	public List<Member> selectAll() {
		// TODO Auto-generated method stub
		return null;
	}

	@Override
	public int update(Member member) {
		// TODO Auto-generated method stub
		return 0;
	}

	@Override
	public int delete(String userId) {
		// TODO Auto-generated method stub
		return 0;
	}

}
```