

PAPAGO TRANSLATION PROJECT

Hyejung(*Jen*) Lim

2020 Feb

TECH SPECS

FRONTEND



BACKEND



SERVER



CLOUD COMPUTING PLATFORM



DATABASE



DEVELOPMENT TOOLS



DEVOPS LIFECYCLE TOOL



DESIGN TOOL



USER MANAGEMENT

Sign Up (*Hashed Passwords*)
& Login (*Interceptor*)
& Logout

01

CONDITIONAL API REQUESTS

Sending an API requests *only when*
it's the *first* request of text

02

SEARCH & DELETION

Searching & Deleting
translated texts

03

TABLE OF CONTENTS

04

SORTING

Sorting *translated* texts
by user & date as well as
by an ascending/descending order

05

DESIGN

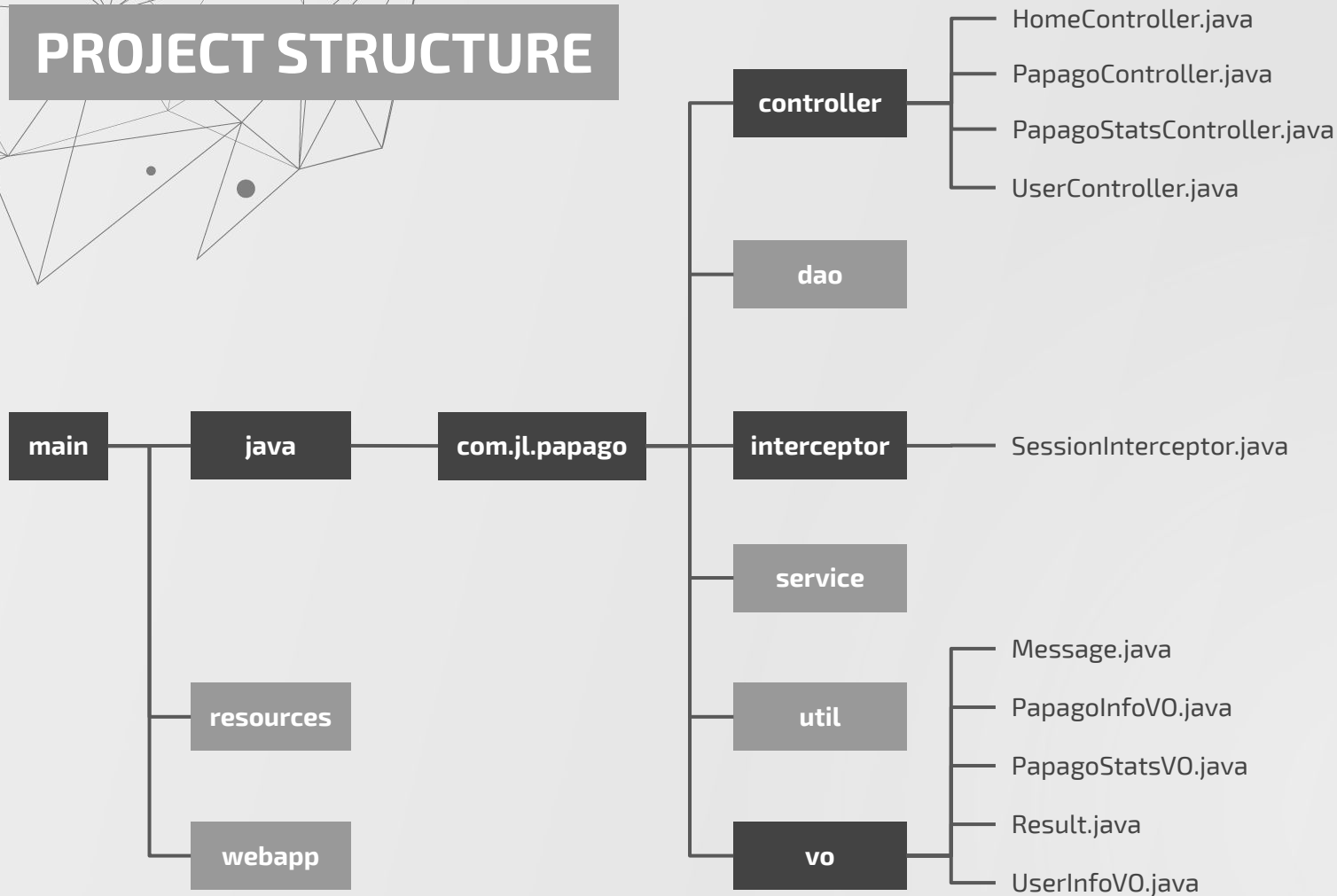
Responsive web design
on *Chrome*

06

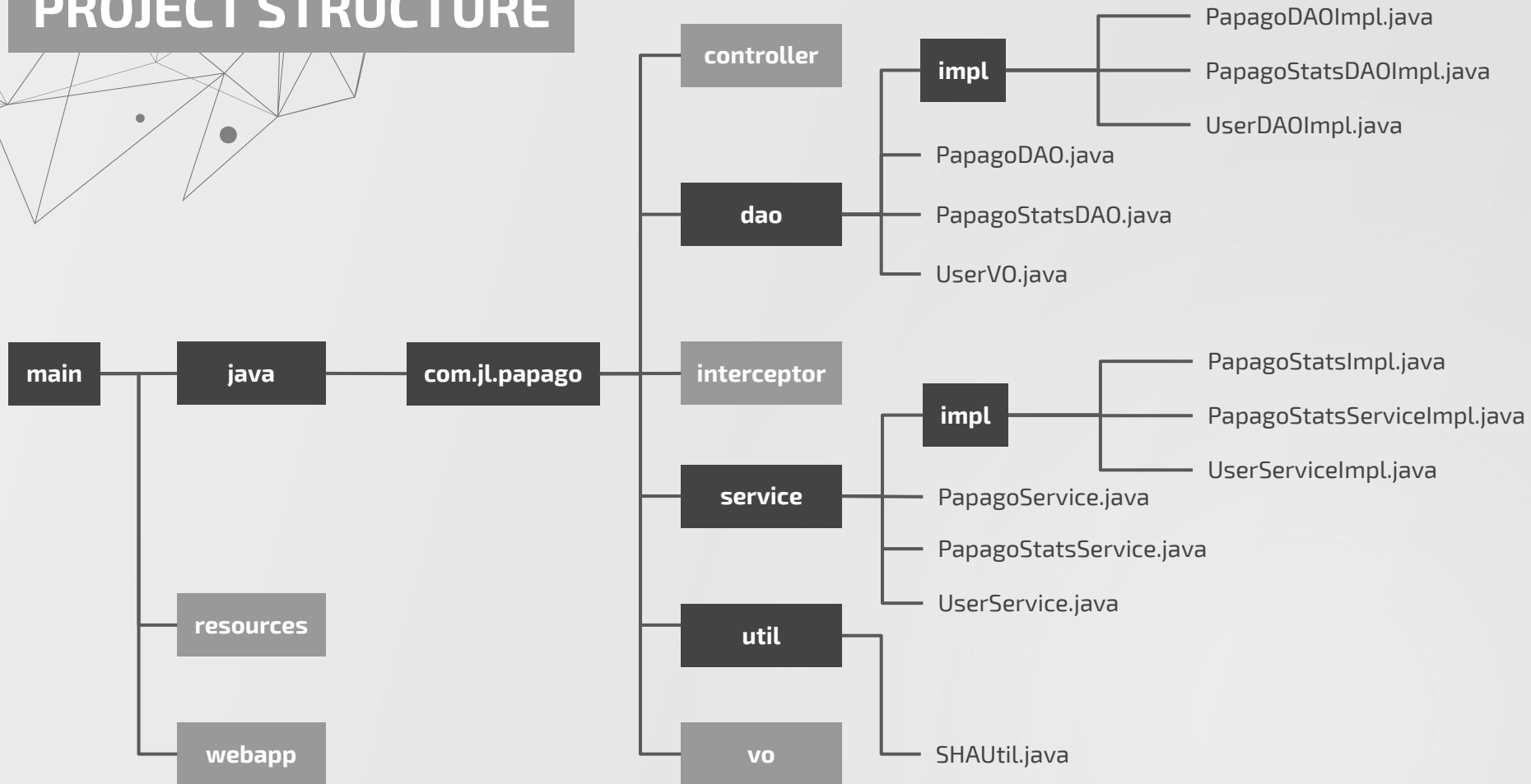
DEPLOYING A WEBSITE

Deploying a website on
AWS EC2 & RDS (*ft. Jenkins*)

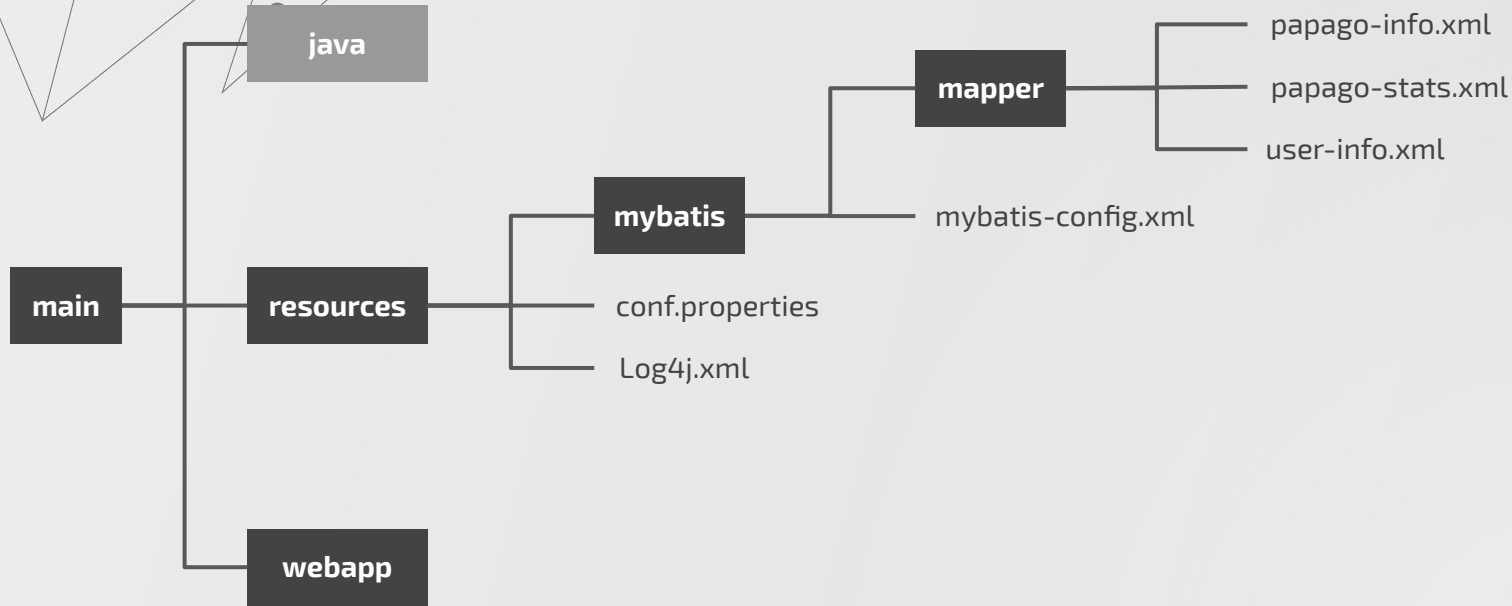
PROJECT STRUCTURE



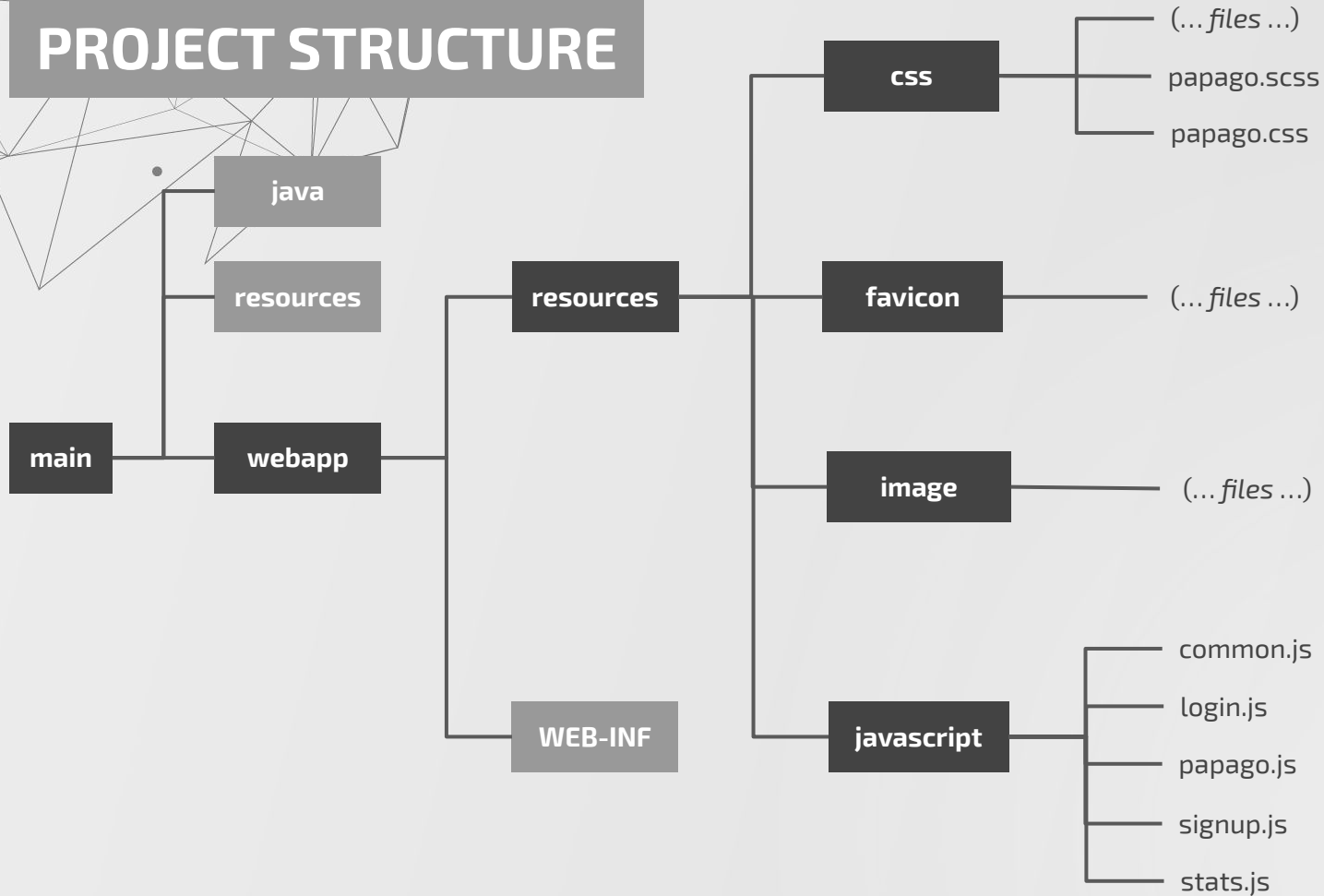
PROJECT STRUCTURE



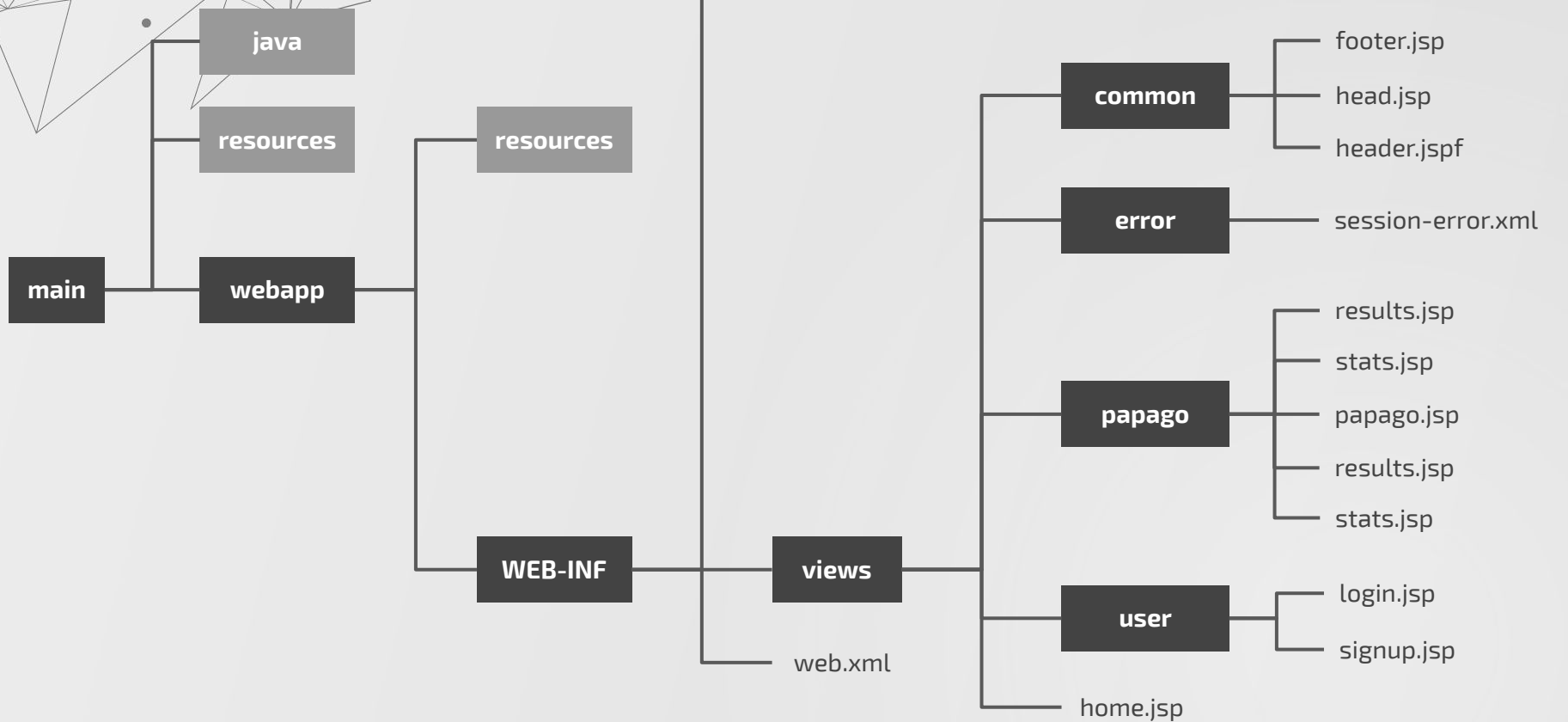
PROJECT STRUCTURE



PROJECT STRUCTURE



PROJECT STRUCTURE



CREATE DATABASE TABLES

```
CREATE TABLE user_info(  
    ui_num INT(11) NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    ui_name VARCHAR(10) NOT NULL,  
    ui_id VARCHAR(10) NOT NULL,  
    ui_pwd VARCHAR(10) NOT NULL,  
    credat CHAR(8) NOT NULL,  
    cretim CHAR(8) NOT NULL,  
    moddat CHAR(8) NOT NULL,  
    modtim CHAR(8) NOT NULL,  
    active CHAR(1) DEFAULT '1'  
) ENGINE=InnoDB DEFAULT CHARSET=UTF8;
```

```
CREATE TABLE papago_info(  
    pi_num INT(11) AUTO_INCREMENT PRIMARY KEY,  
    pi_source char(6) NOT NULL,  
    pi_target char(6) NOT NULL,  
    pi_text VARCHAR(100) NOT NULL,  
    pi_result VARCHAR(300),  
    pi_cnt INT(11) default 1  
) ENGINE=InnoDB DEFAULT CHARSET=UTF8;
```

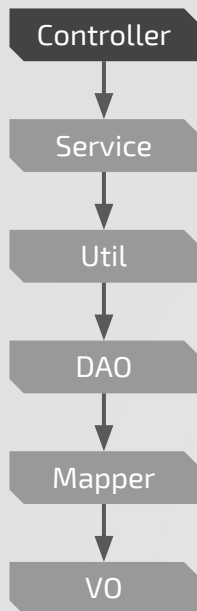
```
CREATE TABLE papago_stats(  
    ps_num INT(11) AUTO_INCREMENT PRIMARY KEY,  
    ui_num INT(11) not null,  
    pi_num INT(11) not null,  
    credat CHAR(8) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=UTF8;
```



01

USER MANAGEMENT

Sign up (*Hashed Passwords*) + Login (*Interceptor*) + Logout



@RestController

```
public class UserController {
```

UserController.java

@Resource

```
private UserService us;
```

@PostMapping("/login")

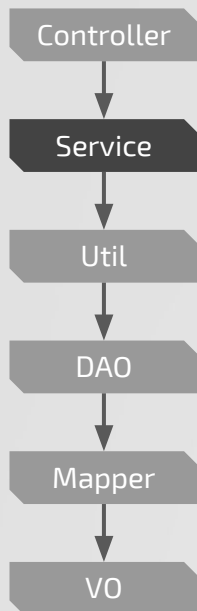
```
public Map<String, Object> doLogin(@RequestBody UserInfoVO ui, HttpSession hs){  
    return us.doLogin(ui, hs);  
}
```

@PostMapping("/logout")

```
public Map<String, String> doLogout(HttpSession hs) {  
    Map<String, String> rMap = new HashMap<>();  
    hs.invalidate();  
    rMap.put("result", "true");  
    return rMap;  
}
```

@PostMapping("/signup")

```
public Map<String, String> doSignup(@RequestBody UserInfoVO ui){  
    return us.doSignup(ui);  
}
```



@Service

```
public class UserServiceImpl implements UserService {
```

UserServiceImpl.java

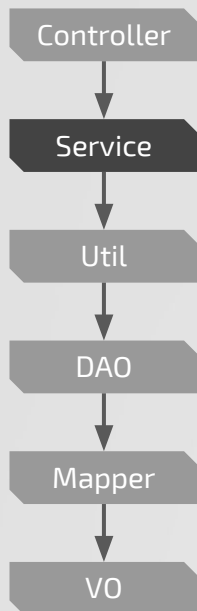
1 of 2

@Resource

```
private UserDao userDao;
```

@Override

```
public Map<String, Object> doLogin(UserInfoVO ui, HttpSession hs) {  
    ui.setUiPwd(SHAUtil.getSHA(ui.getUiPwd()));  
    ui = userDao.doLogin(ui);  
    Map<String, Object> rMap = new HashMap<>();  
    if(ui == null) {  
        rMap.put("msg", "Login Failed !");  
        rMap.put("result", false);  
    } else {  
        rMap.put("msg", "Welcome, " + ui.getUserName());  
        rMap.put("result", true);  
        rMap.put("ui", ui);  
        hs.setAttribute("ui", ui);  
    }  
    return rMap;  
}
```



@Service

```
public class UserServiceImpl implements UserService {
```

@Resource

```
private UserDao udao;
```

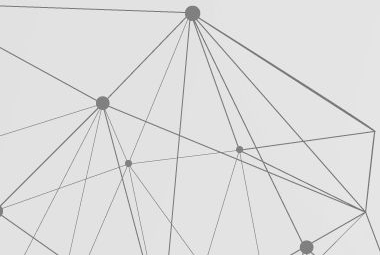
@Override

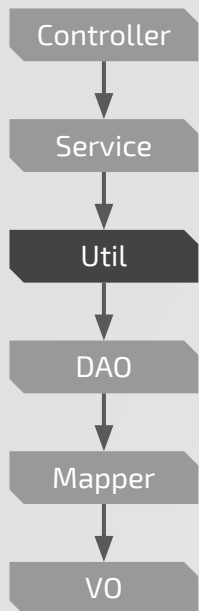
```
public Map<String, String> doSignup(UserInfoVO ui) {  
    Map<String, String> rMap = new HashMap<String, String>();  
    ui.setUiPwd(SHAUtil.getSHA(ui.getUiPwd()));  
    rMap.put("cnt", udao.insertUserInfo(ui) + "");  
    return rMap;  
}
```

```
}
```

ServiceImpl.java

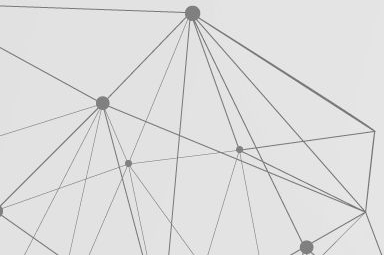
2 of 2

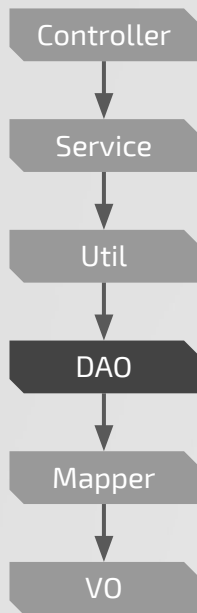




```
public class SHAUtil {  
    private final static String SALT_KEY="(write a phrase)";  
  
    public static String getSHA(String source) {  
        String target = Digest.sha256Hex(source+SALT_KEY);  
        return target;  
    }  
}
```

SHAUtil.java





@Repository

```
public class UserDAOImpl implements UserDAO {
```

@Resource

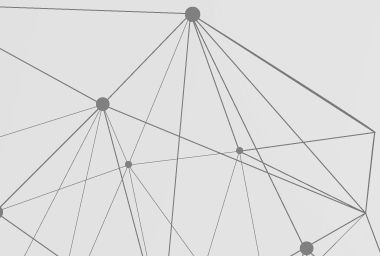
```
private SqlSessionFactory ssf;
```

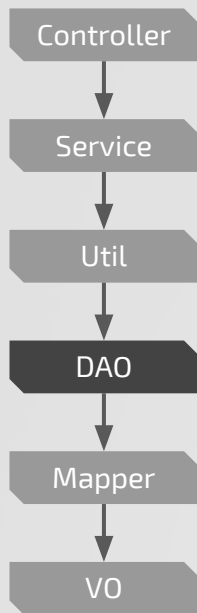
@Override

```
public UserInfoVO doLogin(UserInfoVO user) {  
    SqlSession ss = ssf.openSession();  
    try {  
        return ss.selectOne("com.jl.papago.dao.UserInfoMapper.doLogin", user);  
    } catch (Exception e) {  
        log.error(e.getMessage());  
    } finally {  
        ss.close();  
    }  
    return null;  
}  
}
```

UserDAOImpl.java

1 of 2





@Repository

```
public class UserDAOImpl implements UserDAO {
```

@Resource

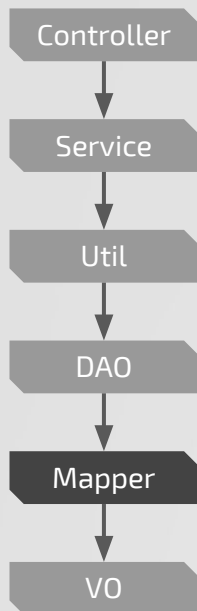
```
private SqlSessionFactory ssf;
```

@Override

```
public int insertUserInfo(UserInfoVO user) {  
    SqlSession ss = ssf.openSession();  
    try {  
        int cnt = ss.insert("com.jl.papago.dao.UserInfoMapper.insertUserInfo", user);  
        ss.commit();  
        return cnt;  
    } catch (Exception e) {  
        e.printStackTrace();  
    } finally {  
        ss.close();  
    }  
    return 0;  
}  
}
```

UserDAOImpl.java

2 of 2



```
<mapper namespace="com.jl.papago.dao.UserInfoMapper">
```

```
<select id="doLogin" resultType="user">
```

```
    select * from user_info where ui_id=#{uiId} and ui_pwd=#{uiPwd}
```

```
</select>
```

```
<insert id="insertUserInfo" parameterType="user">
```

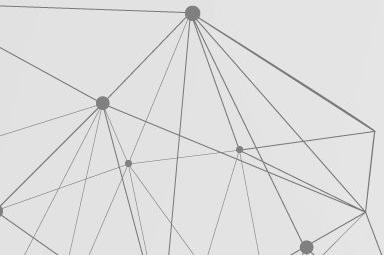
```
    insert into user_info(ui_id, ui_pwd, ui_name, credat, cretim, moddat, modtim)
```

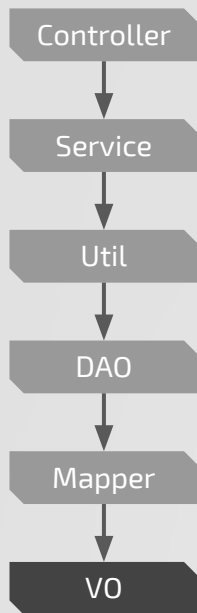
```
    values(#{uiId}, #{uiPwd}, #{uiName}, date_format(NOW(), '%Y%m%d'),  
           date_format(NOW(), '%H%i%S'), date_format(NOW(), '%Y%m%d'),  
           date_format(NOW(), '%H%i%S'))
```

```
</insert>
```

```
</mapper>
```

user-info.xml

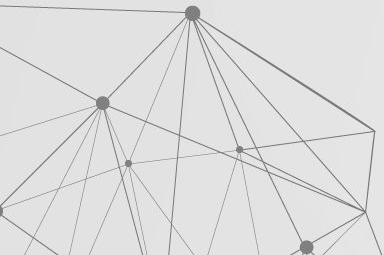




@Data

```
public class UserInfoVO {  
    private Integer uiNum;  
    private String uiName;  
    private String active;  
    private String uild;  
    private String uiPwd;  
    private String moddat;  
    private String modtim;  
    private String credat;  
    private String cretim;  
}
```

UserInfoVO.java



SIGN UP PAGE

Create Your Account

Name

ID


Password

Password Check

Sign Up

Index Page

Your password is stored in a hashed form.




ERROR

An ID should contain more than 5 characters !

OK

Sign Up

Index Page



Congratulations!

You successfully signed up.

OK

Sign Up

Index Page

```
function insertUser(){
  var xhr = new XMLHttpRequest();
  xhr.open('POST', '/signup');
  xhr.setRequestHeader('Content-Type', 'application/json');
  xhr.onreadystatechange = function(){
    if(xhr.readyState == 4 && xhr.status == 200){
      var res = JSON.parse(xhr.responseText);
      if(res.cnt == 1){
        Swal.fire({
          position: 'top',
          icon: 'success',
          title: 'Congratulations!',
          text: 'You successfully signed up.'
        }).then(() => {
          location.href = '/views/user/login';
        })
      }
    }
  }
}
```

```
var param = {
  uiName : document.getElementById('signup-uiName').value,
  uid : document.getElementById('signup-uid').value,
  uiPwd : document.getElementById('signup-uiPwd').value
}
param = JSON.stringify(param);
xhr.send(param);
```

signup.js

```
function checkForm() {  
    var uiName = document.getElementById("signup-uiName");  
    var uild = document.getElementById("signup-uild");  
    var uiPwd = document.getElementById("signup-uiPwd");  
    var uiPwdCheck = document.getElementById("signup-uiPwdCheck");
```

```
    if(uiName.value.trim().length < 2){  
        (code for an alert message)  
        uiName.value = "";  
        uiName.focus();  
        return false;  
    }
```

```
    if(uild.value.trim().length < 5){  
        (code for an alert message)  
        uild.value = "";  
        uild.focus();  
        return false;  
    }
```

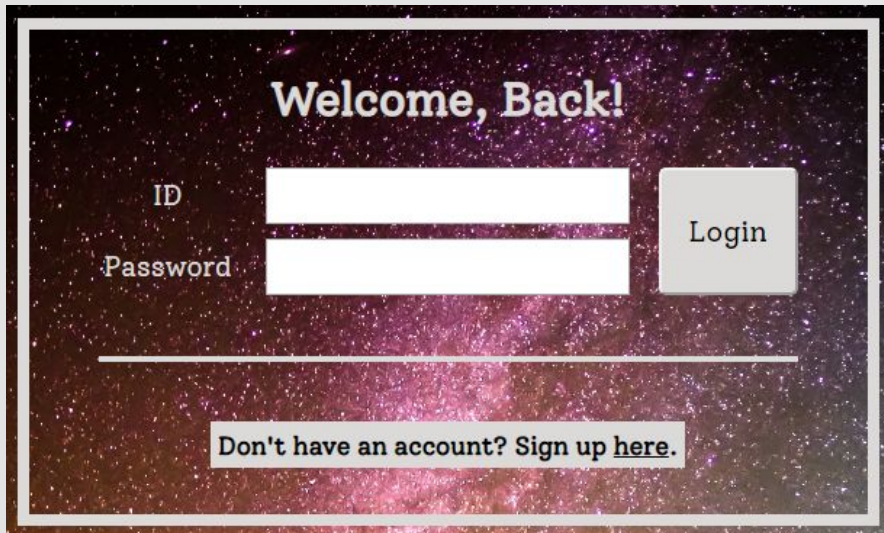
```
    if(uiPwd.value.trim().length < 5){  
        (code for an alert message)  
        uiPwd.value = "";  
        uiPwd.focus();  
        return false;  
    }
```

```
    if(uiPwd.value != uiPwdCheck.value){  
        (code for an alert message)  
        uiPwdCheck.value = "";  
        uiPwdCheck.focus();  
        return false;  
    }
```

```
    insertUser();  
}
```

signup.js

LOGIN PAGE



The login form is set against a dark space background with a vibrant pink and purple nebula. It features a white border and a white title. Below the title are two white input fields for ID and Password, followed by a grey Login button. A horizontal line separates the login fields from a link at the bottom.

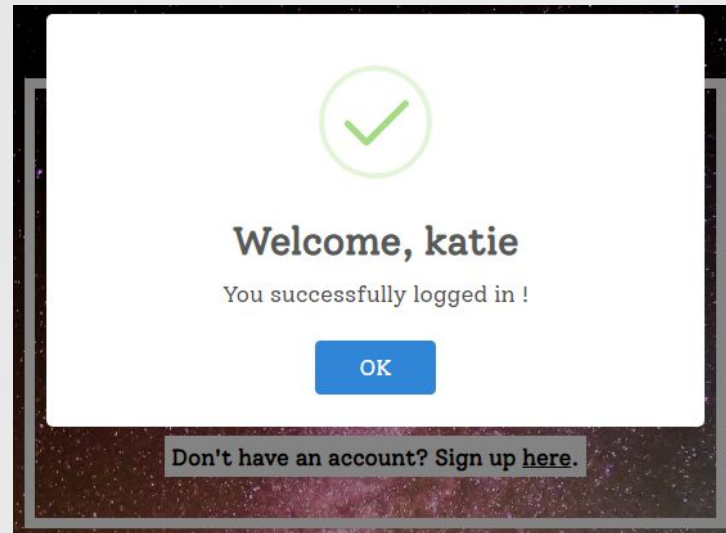
Welcome, Back!

ID


Password

Login

Don't have an account? Sign up [here.](#)



The confirmation screen has a white background with a green checkmark icon. It displays a personalized welcome message and a success message. A blue OK button is centered below the message. The bottom link is on a grey background.



Welcome, katie

You successfully logged in !

OK

Don't have an account? Sign up [here.](#)

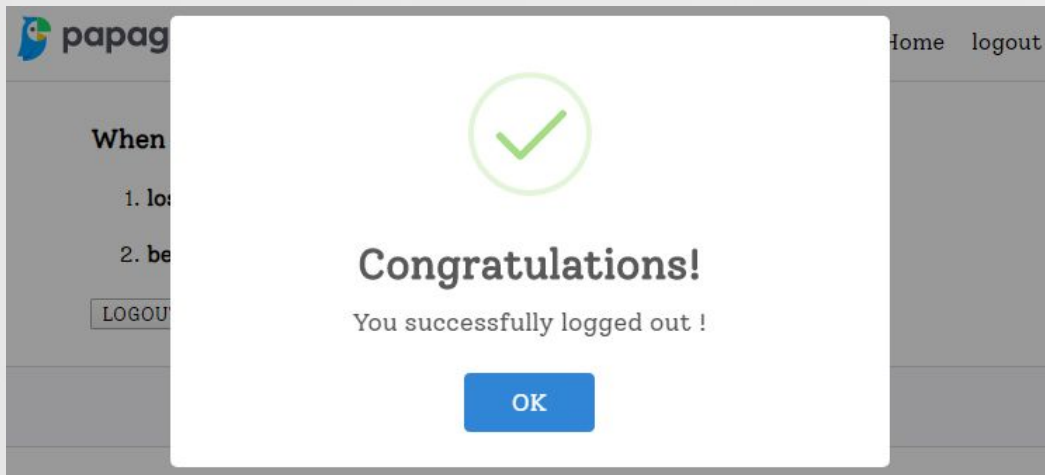
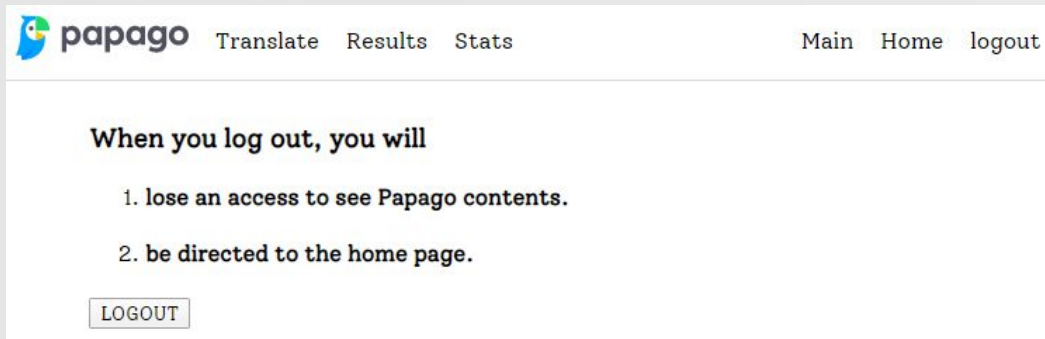
```
$(document).ready(function(){
    $('button').on('click', function(){
        var param = {
            uid: $('#login-uid').val(),
            uiPwd: $('#login-uiPwd').val()
        }
        param = JSON.stringify(param);
        $.ajax({
            url: '/login',
            method: 'POST',
            data: param,
            beforeSend: function(xhr){
                xhr.setRequestHeader('x-ajax', true);
            },
            contentType: 'application/json',
            success: function(res){
                (code for an alert message)
            },
            error: function(res){
                console.log(res);
            }
        });
    });
});
```

login.js

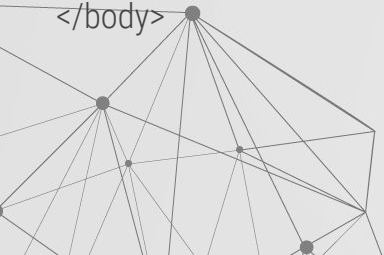
```
var signupText = document.getElementById('move-to-signup');

signupText.addEventListener('click', function(){
    location.href="/views/user/signup";
}, false);
```

LOGOUT PAGE




```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>LOGOUT</title>
<jsp:include page="/WEB-INF/views/common/head.jsp"></jsp:include>
</head>
<body>
<div id="result">
    <h3>When you log out, you will</h3>
    <ol>
        <li><h4>lose an access to see Papago contents.</h4></li>
        <li><h4>be directed to the home page.</h4></li>
    </ol>
    <button class="index-buttons"><span>LOGOUT</span></button>
</div>
</body>
```



```
<script>
$(document).ready(function(){
    $('button').on('click', function(){
        $.ajax({
            url: '/logout',
            method: 'POST',
            success: function(res){
                if(res.result == 'true'){
                    Swal.fire({
                        (code)
                    }).then(() => {
                        location.href = '/';
                    })
                }
            },
            error: function(res){
                console.log(res);
            }
        });
    });
});
</script>
</html>
```

logout.js

ERROR PAGE

Wait...!

You're trying to access to user-only contents.

Please log in to continue.

Login

```
<%@ include file = "/WEB-INF/views/common/header.jspf" %>
```

```
<link href="{cssPath}/papago.css" rel="stylesheet" />
```

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/animate.css/3.7.2/animate.min.css">
```

```
<link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
```

```
<link href="https://fonts.googleapis.com/css?family=Solway&display=swap" rel="stylesheet">
```

```
<!-- FAVICON START -->
```

```
<link rel="icon" type="image/ico" href="{faviconPath}/favicon.ico"/>
```

```
<link rel="apple-touch-icon" sizes="180x180" href="{faviconPath}/apple-touch-icon.png">
```

```
<link rel="icon" type="image/png" sizes="32x32" href="{faviconPath}/favicon-32x32.png">
```

```
<link rel="icon" type="image/png" sizes="16x16" href="{faviconPath}/favicon-16x16.png">
```

```
<link rel="manifest" href="{faviconPath}/site.webmanifest">
```

```
<link rel="mask-icon" href="{faviconPath}/safari-pinned-tab.svg" color="#5bbad5">
```

```
<meta name="msapplication-TileColor" content="#da532c">
```

```
<meta name="theme-color" content="#ffffff">
```

```
<!-- FAVICON END -->
```

```
<script src="{jsPath}/common.js"></script>
```

```
<script src="https://kit.fontawesome.com/292ac90a40.js" crossorigin="anonymous"></script>
```

```
<script src="https://code.jquery.com/jquery-3.4.1.min.js"
```

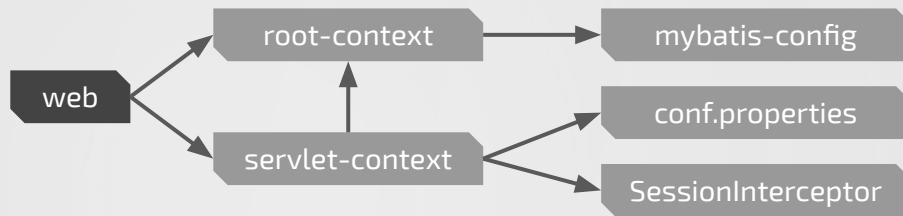
```
    integrity="sha256-CSXorXvZcTkaix6Yo6HppcZGetbYMGWSFIBw8HfCJo="
    crossorigin="anonymous"></script>
```

head.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8" trimDirectiveWhitespaces="true" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions" %>
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<c:set var="cssPath" value="${pageContext.request.contextPath}/resources/css" />
<c:set var="faviconPath" value="${pageContext.request.contextPath}/resources/fv" />
<c:set var="jsPath" value="${pageContext.request.contextPath}/resources/js" />
<c:set var="prismPath" value="${pageContext.request.contextPath}/resources/prism" />
```



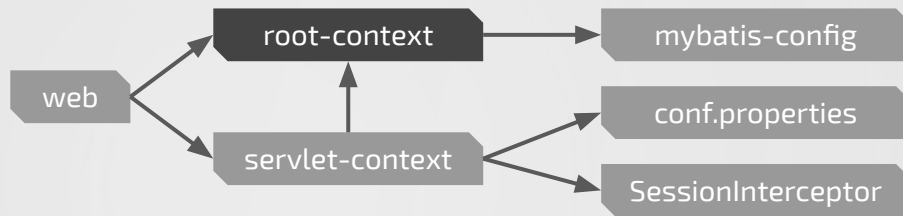
web.xml

<!-- The definition of the Root Spring Container shared by all Servlets and Filters -->

```
<context-param>
  <param-name>contextConfigLocation</param-name>
  <param-value>/WEB-INF/spring/root-context.xml</param-value>
</context-param>
```

<!-- Processes application requests -->

```
<servlet>
  <servlet-name>appServlet</servlet-name>
  <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
  <init-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>/WEB-INF/spring/appServlet/servlet-context.xml</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>
```



root-context.xml

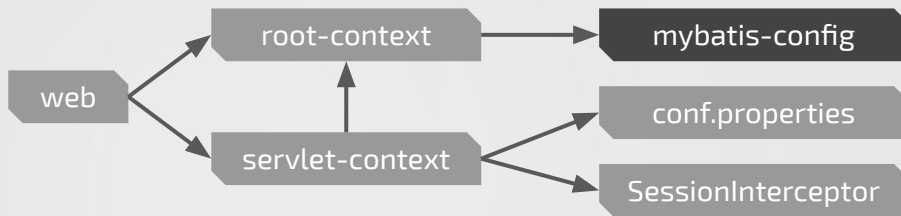
<context-param>

<!-- Root Context: defines shared resources visible to all other web components -->

```
<bean id="bds" class="org.apache.commons.dbcp.BasicDataSource" destroy-method="close">
  <property name="driverClassName" value="org.mariadb.jdbc.Driver" />
  <property name="url" value="jdbc:mariadb://(DB Address):3003/innodb" />
  <property name="defaultAutoCommit" value="false" />
  <property name="username" value="(Username)" />
  <property name="password" value="(Password)" />
</bean>
```

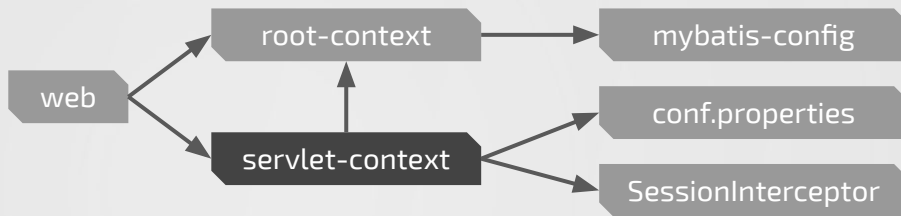
<!-- Creates memory -->

```
<bean id="ssfb" class="org.mybatis.spring.SqlSessionFactoryBean">
  <property name="dataSource" ref="pds" />
  <property name="configLocation" value="classpath:/mybatis/mybatis-config.xml"></property>
</bean>
```



mybatis-config.xml

```
<settings>
  <setting name="mapUnderscoreToCamelCase" value="true" />
</settings>
<typeAliases>
  <typeAlias type="com.jl.papago.vo.PapagoInfoVO" alias="tvo"/>
  <typeAlias type="com.jl.papago.vo.UserInfoVO" alias="user"/>
  <typeAlias type="com.jl.papago.vo.PapagoStatsVO" alias="ps"/>
</typeAliases>
<mappers>
  <mapper resource="mybatis/mapper/papago-info.xml"/>
  <mapper resource="mybatis/mapper/user-info.xml"/>
  <mapper resource="mybatis/mapper/papago-stats.xml"/>
</mappers>
```



servlet-context.xml

1 of 2

<!-- DispatcherServlet Context: defines this servlet's request-processing infrastructure -->

```
<beans:bean class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">
    <beans:property name="location" value="classpath:/conf.properties" />
</beans:bean>
```

<!-- Enables the Spring MVC @Controller programming model -->

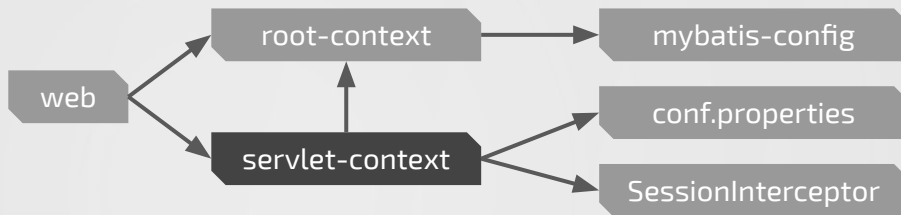
```
<mvc:annotation-driven />
```

*<!-- Handles HTTP GET requests for /resources/** by efficiently serving up static resources -->*

```
<mvc:resources mapping="/resources/**" location="/resources/" />
```

<!-- Resolves views selected for rendering by @Controllers to .jsp resources in the /WEB-INF/views directory -->

```
<beans:bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
    <beans:property name="prefix" value="${prefix.value}" />
    <beans:property name="suffix" value=".jsp" />
</beans:bean>
```

servlet-context.xml

2 of 2

<!-- Reads all annotations & looks for root-context.xml (It's possible since 'ssfb' was already created.) -->

```
<context:component-scan base-package="com.jl.papago" />
```

```
<mvc:interceptors>
```

```
  <mvc:interceptor>
```

```
    <mvc:mapping path="/views/**" />
```

```
    <mvc:exclude-mapping path="/" />
```

```
    <mvc:exclude-mapping path="/views/user/login" />
```

```
    <mvc:exclude-mapping path="/login" />
```

```
    <mvc:exclude-mapping path="/views/user/signup" />
```

```
    <mvc:exclude-mapping path="/signup" />
```

```
    <mvc:exclude-mapping path="/views/error/*" />
```

```
    <beans:ref bean="sessionInterceptor" />
```

```
  </mvc:interceptor>
```

```
</mvc:interceptors>
```

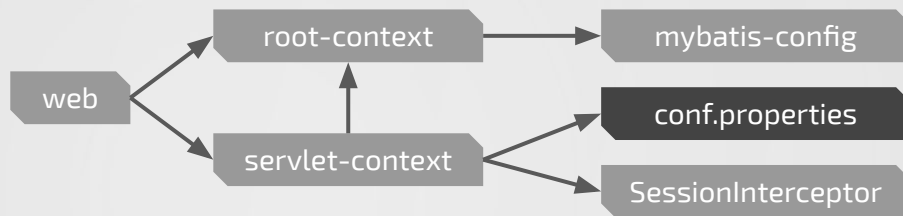
<!-- login page -->

<!-- login logic -->

<!-- signup page -->

<!-- signup logic -->

<!-- error page -->



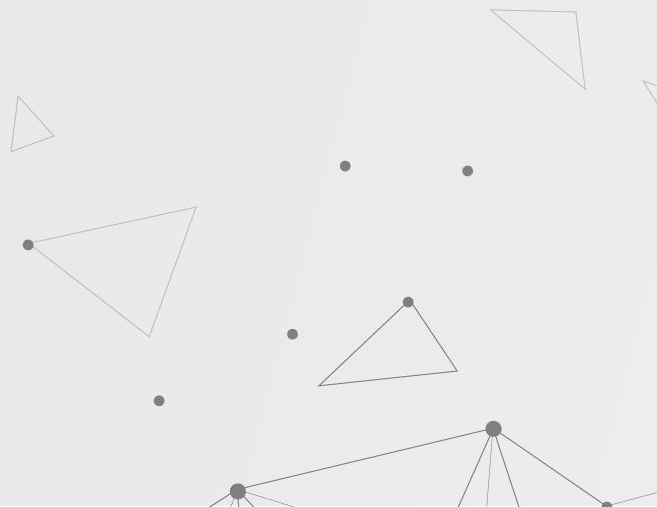
conf.properties

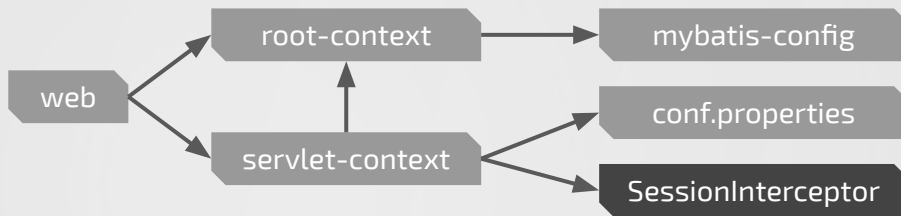
client.id=(Papago API Client Id #)

client.secret=(Papago API Client Secret #)

client.api.url=https://openapi.naver.com/v1/papago/n2mt

prefix.value=/WEB-INF/views/





SessionInterceptor.java

1 of 2

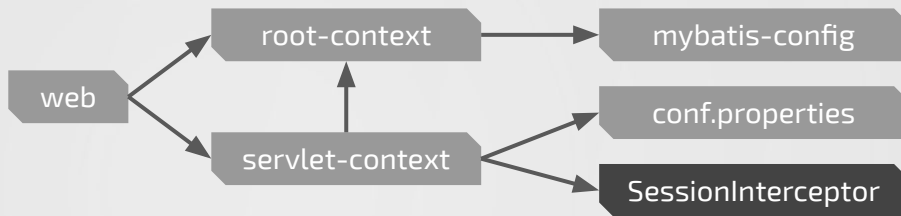
@Component

```
public class SessionInterceptor extends HandlerInterceptorAdapter{
```

```
    private ObjectMapper om = new ObjectMapper();
```

@Override

```
    public boolean preHandle(HttpServletRequest req, HttpServletResponse res, Object handler) throws IOException {  
        String uri = req.getRequestURI();  
        HttpSession hs = req.getSession();
```



SessionInterceptor.java

2 of 2

```
if(hs.getAttribute("ui") == null) {  
    String isAjax = req.getHeader("x-ajax");  
    if(isAjax != null) {  
        Map<String, String> map = new HashMap<>();  
        map.put("msg", "Login FAILED!!!");  
        res.setContentType("application/json;charset=utf-8");  
        PrintWriter pw = res.getWriter();  
        pw.write(om.writeValueAsString(map));  
        return false;  
    }  
    res.sendRedirect("/views/error/session-error");  
    return false;  
}  
return true;  
}
```



02

Conditional API Requests

Sending an API request *only when* it's the *first* request of text

Papago API Response

Request

Method

Request URL

POST

https://openapi.naver.com/v1/papago/n2mt

SEND

Parameters

Headers

Body

Header name

Header value

Content-Type

application/x-www-form-urlencoded; charset=UTF-8

Header name

Header value

X-Naver-Client-Id

(YOUR CLIENT ID)

Header name

Header value

X-Naver-Client-Secret

(YOUR CLIENT SECRET)

200 OK 96.59 ms

```
{
  - "message": {
    "@type": "response",
    "@service": "naverservice.nmt.proxy",
    "@version": "1.0.0",
    - "result": {
      "srcLangType": "en",
      "tarLangType": "ko",
      "translatedText": "번역."
    }
  }
}
```

Headers

Body

Body content type

Editor view

application/x-www-form-urlencoded

Form data (www-url-form-encoded)

ENCODE PAYLOAD

DECODE PAYLOAD

source

en

target

ko

text

translation

(Tested with Google App "Advanced REST client")



English ▼

translation

Korean ▼

번역.

Translate

Korean

- English, Japanese, Chinese(Simplified), Chinese(Traditional), Spanish, French, German, Russian, Italian, Vietnamese, Thai, Indonesian

Chinese(Simplified)

- Chinese(Traditional), Japanese

Chinese(Traditional)

- Japanese

English

- Japanese, Chinese(Simplified), Chinese(Traditional), French

Translation Results

Papago Results !

Source Lang ☐

Target Lang ☐

Source Text ☒

Target Text ☐

SELECT ALL ☐

trans

SEARCH

DELETE

Delete	Record # ▼	Source Lang ▼	Target Lang ▼	Source Text ▼	Target Text ▼	Count ▼
<input type="checkbox"/>	945	en	ko	translation	번역.	3

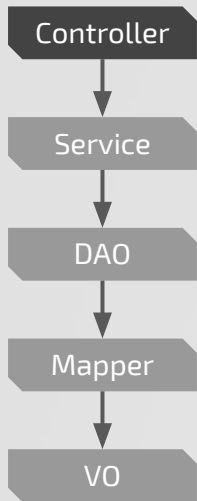
Stats list

By Date

By User

All

Record # ▼	Request User # ▼	Request User ID ▼	Translation # ▼	Date ▼
1687	26	summer	945	20200202
1686	26	summer	945	20200202
1685	26	summer	945	20200202



@RestController

```
public class PapagoController {
```

@Resource

```
private PapagoService ps;
```

@PostMapping("/papago")

```
public Message doTranslate(@RequestBody TransVO tvo, HttpSession hs) {
```

```
    UserInfoVO ui = (UserInfoVO)hs.getAttribute("ui");
```

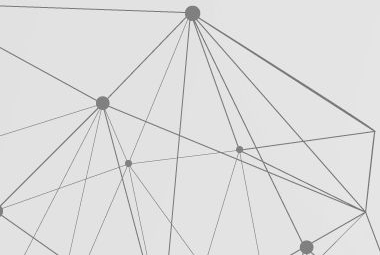
```
    tvo.setUiNum(ui.getUiNum());
```

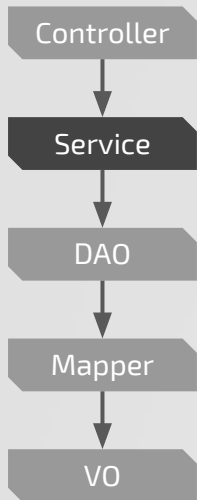
```
    return ps.doTranslate(tvo);
```

```
}
```

```
}
```

PapagoController.java





`@Service("papagoService")`

`public class PapagoServiceImpl implements PapagoService {`

`@Value("${client.id}")`

`private String id;`

`@Value("${client.secret}")`

`private String secret;`

`@Value("${client.api.url}")`

`private String apiUrl;`

`@Resource`

`private PapagoDAO pdao;`

`@Resource`

`private PapagoStatsDAO psdao;`

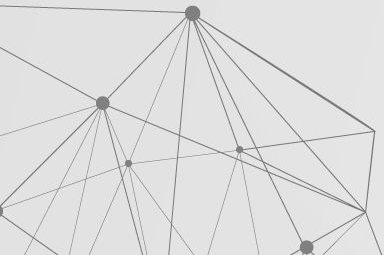
`// Ignore data that doesn't follow the rules of Papago API`

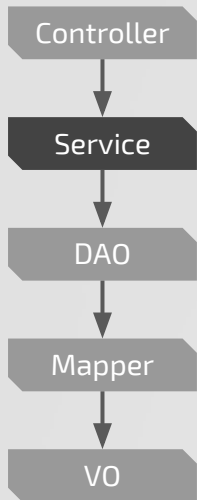
`private ObjectMapper om = new ObjectMapper()`

`.configure(DeserializationFeature.FAIL_ON_UNKNOWN_PROPERTIES, false);`

PapagoServiceImpl.java

1 of 5





@Override

```
public Message doTranslate(TransVO tvo) {  
    try {
```

```
        PapagoStatsVO ps = new PapagoStatsVO();  
        ps.setUiNum(tvo.getUiNum());
```

```
        PapagoInfoVO pi = new PapagoInfoVO();  
        pi.setPiSource(tvo.getSource());  
        pi.setPiTarget(tvo.getTarget());  
        pi.setPiText(tvo.getText());  
        pi = pdao.selectPapagoInfo(pi);
```

```
        if(pi != null) {  
            Result r = new Result();  
            r.setSrcLangType(pi.getPiSource());  
            r.setTarLangType(pi.getPiTarget());  
            r.setTranslatedText(pi.getPiResult());  
            pdao.updatePapagoInfoForCnt(pi);  
            ps.setPiNum(pi.getPiNum());  
            psdao.insertPapagoStats(ps);  
            Message m = new Message();  
            m.setResult(r);  
            return m;  
        }
```

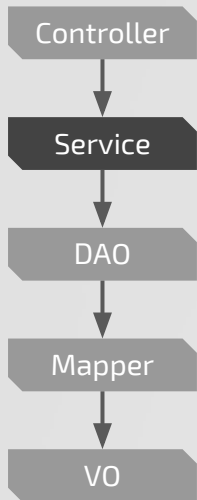
PapagoServiceImpl.java

2 of 5

If the requested text had **already** been translated in the past

Increase the count for the translated text

Add info of the translated text to PapagoStatsMapper



<!-- *** Sending an API request START *** -->

```
String text = URLEncoder.encode(tvo.getText(), "UTF-8");
```

```
URL url = new URL(apiUrl);
```

```
HttpURLConnection hc = (HttpURLConnection)url.openConnection();
```

```
hc.setRequestMethod("POST");
```

```
hc.setRequestProperty("Content-Type", "application/x-www-form-urlencoded; charset=UTF-8");
```

```
hc.setRequestProperty("X-Naver-Client-Id", id);
```

```
hc.setRequestProperty("X-Naver-Client-Secret", secret);
```

```
String param = "source=" + tvo.getSource() + "&target=" + tvo.getTarget() + "&text=" + text;
```

```
hc.setDoOutput(true);
```

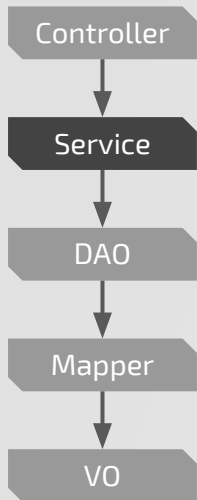
```
DataOutputStream dos = new DataOutputStream(hc.getOutputStream());
```

```
dos.writeBytes(param);
```

```
dos.flush();
```

```
dos.close();
```

<!-- *** Sending an API request END *** -->



<!-- *** Getting a response of the API request START *** -->

```
int status = hc.getResponseCode();
```

```
InputStreamReader isr = new InputStreamReader(hc.getInputStream());
```

```
BufferedReader br = new BufferedReader(isr);
```

```
StringBuffer res = new StringBuffer();
```

```
String str = null;
```

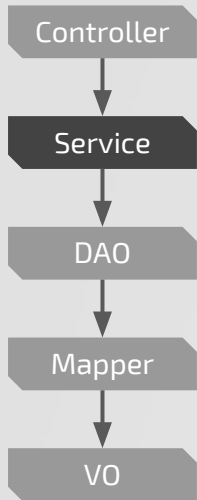
```
while((str=br.readLine())!=null) {
```

```
    res.append(str);
```

```
}
```

```
br.close();
```

```
TransVO resultTvo = om.readValue(res.toString(), TransVO.class);
```

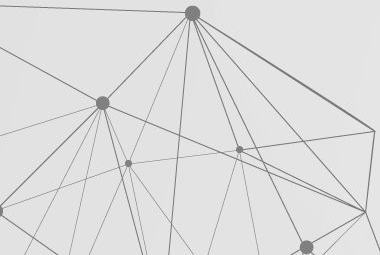


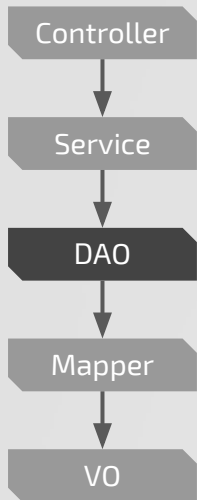
```
Result r = resultTvo.getMessage().getResult();
pi = new PapagoInfoVO();
pi.setPiSource(r.getSrcLangType());
pi.setPiTarget(r.getTarLangType());
pi.setPiResult(r.getTranslatedText());
pi.setPiText(tvo.getText());
pdao.insertPapagoInfo(pi);

ps.setPiNum(pi.getPiNum());
psdao.insertPapagoStats(ps);
return resultTvo.getMessage();
<!-- *** Getting a response of the API request END *** -->
```

```
} catch (UnsupportedEncodingException e) {
    e.printStackTrace();
} catch (MalformedURLException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}
return null;
```

```
}
```





@Repository

```
public class PapagoDAOImpl implements PapagoDAO {
```

@Resource

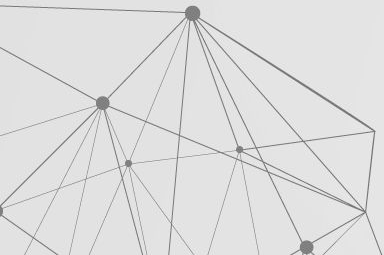
```
private SqlSessionFactory ssf;
```

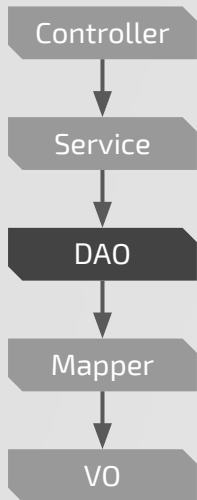
@Override

```
public PapagoInfoVO selectPapagoInfo(PapagoInfoVO tvo) {  
    SqlSession ss = ssf.openSession();  
    try {  
        return ss.selectOne("com.jl.papago.dao.PapagoInfoMapper.selectPapagoInfo", tvo);  
    } catch (Exception e) {  
        e.printStackTrace();  
    } finally {  
        ss.close();  
    }  
    return null;  
}
```

PapagoDAOImpl.java

1 of 2





@Override

```
public int insertPapagoInfo(PapagoInfoVO tvo) {
```

```
    SqlSession ss = ssf.openSession();
```

```
    try {
```

```
        int cnt = ss.insert("com.jl.papago.dao.PapagoInfoMapper.insertPapagoInfo", tvo);
```

```
        ss.commit();
```

```
        return cnt;
```

```
    } catch (Exception e) {
```

```
        e.printStackTrace();
```

```
    } finally {
```

```
        ss.close();
```

```
    }
```

```
    return 0;
```

```
}
```

@Override

```
public int updatePapagoInfoForCnt(PapagoInfoVO tvo) {
```

```
    SqlSession ss = ssf.openSession();
```

```
    try {
```

```
        int cnt = ss.update("com.jl.papago.dao.PapagoInfoMapper.updatePapagoInfoForCnt", tvo);
```

```
        (same as above)
```

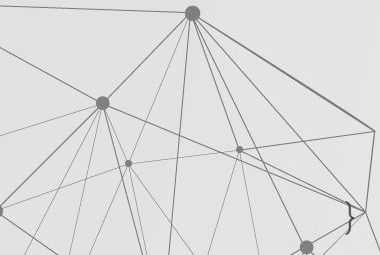
```
    }
```

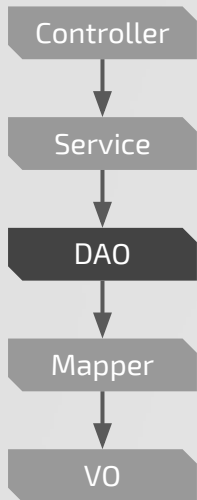
```
    return 0;
```

```
}
```

PapagoDAOImpl.java

2 of 2





@Repository

```
public class PapagoStatsDAOImpl implements PapagoStatsDAO {
```

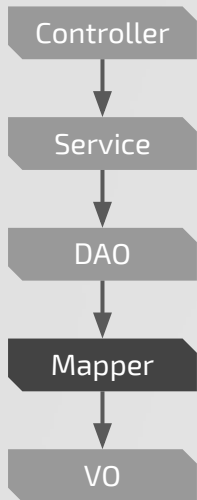
PapagoStatsDAOImpl.java

@Resource

```
private SqlSessionFactory ssf;
```

@Override

```
public int insertPapagoStats(PapagoStatsVO ps) {  
    SqlSession ss = ssf.openSession(false);  
    try {  
        int cnt = ss.insert("com.jl.papago.dao.PapagoStatsMapper.insertPapagoStats", ps);  
        ss.commit();  
        return cnt;  
    } catch (Exception e) {  
        ss.rollback();  
        log.error(e.getMessage());  
    } finally {  
        ss.close();  
    }  
    return 0;  
}
```

```
<mapper namespace="com.jl.papago.dao.PapagoInfoMapper">
```

```
  <select id="selectPapagoInfo" parameterType="tvo" resultType="tvo">
```

```
    select * from papago_info
```

```
    where
```

```
      pi_target = rpad(#{piTarget}, 6, '')
```

```
      and pi_source = rpad(#{piSource}, 6, '')
```

```
      and pi_text = #{piText}
```

```
  </select>
```

```
  <insert id="insertPapagoInfo" useGeneratedKeys="true" keyProperty="piNum" keyColumn="pi_num">
```

```
    insert into papago_info(pi_source, pi_target, pi_text, pi_result)
```

```
    values(#{piSource}, #{piTarget}, #{piText}, #{piResult})
```

```
  </insert>
```

```
  <update id="updatePapagoInfoForCnt">
```

```
    update papago_info
```

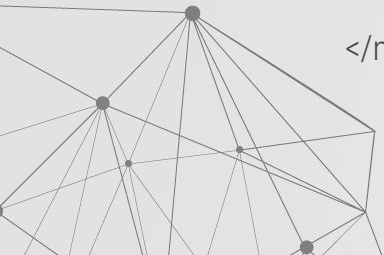
```
    set pi_cnt = pi_cnt + 1
```

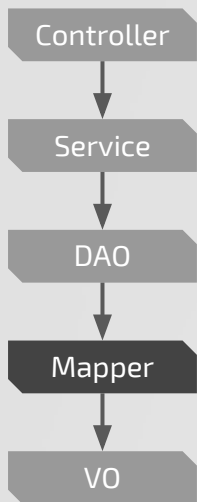
```
    where pi_num = #{piNum}
```

```
  </update>
```

```
</mapper>
```

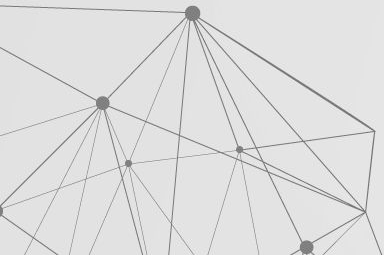
papago-info.xml

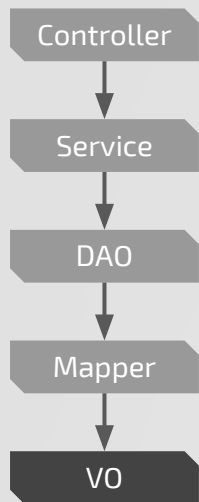




```
<mapper namespace="com.jl.papago.dao.PapagoStatsMapper">
  <insert id="insertPapagoStats" parameterType="ps">
    insert into papago_stats(ui_num, pi_num, credat)
    values(#{uiNum}, #{piNum}, date_format(NOW(), '%Y%m%d'))
  </insert>
</mapper>
```

papago-stats.xml





@Data

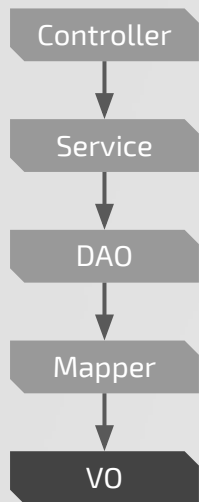
```
public class PapagoInfoVO {  
    private Integer piNum;  
    private String piSource;  
    private String piTarget;  
    private String piText;  
    private String piResult;  
    private Integer piCnt;  
}
```

@Data

```
public class PapagoStatsVO {  
    private Integer psNum;  
    private Integer uiNum;  
    private Integer piNum;  
    private String credat;  
    private String uild;  
    private Integer cnt;  
}
```

PapagoInfoVO.java

PapagoStatsVO.java



@Data

@JsonIgnoreProperties(ignoreUnknown = true)

```
public class TransVO {  
    private String source;  
    private String target;  
    private String text;  
    private Message message;  
    private Integer uiNum;  
}
```

@Data

```
public class Message {  
    private Result result;  
}
```

@Data

```
public class Result {  
    private String srcLangType;  
    private String tarLangType;  
    private String translatedText;  
}
```

TransVO.java

Message.java

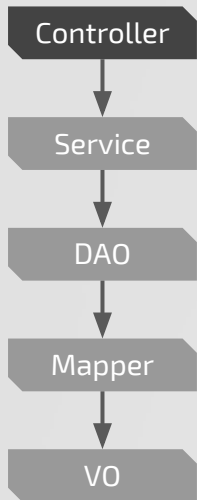
Result.java



03

SEARCH & DELETION

Searching & Deleting translated texts



@RestController

```
public class PapagoController {
```

@Resource

```
private PapagoService ps;
```

@GetMapping("/papago")

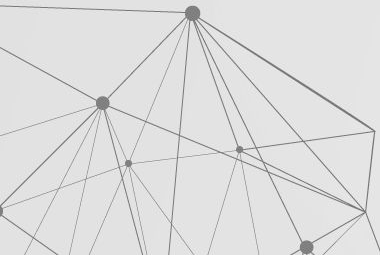
```
public List<PapagoInfoVO> getPapagoInfo(@ModelAttribute PapagoInfoVO param){  
    return ps.getPapagoVOList(param);  
}
```

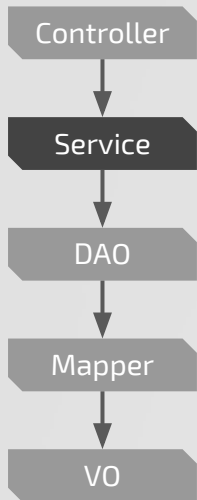
@DeleteMapping("/papago")

```
public Map<String, String> deleteUser(@ModelAttribute PapagoInfoVO param){  
    return ps.deletePapagoInfo(param);  
}
```

```
}
```

PapagoController.java





@Service("papagoService")

public class **PapagoServiceImpl** implements **PapagoService** {

@Resource

private **PapagoDAO** *pdo*;

@Override

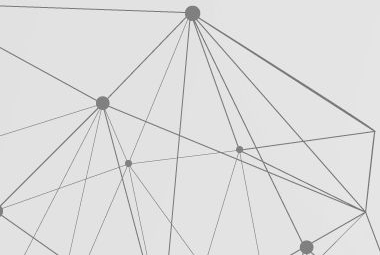
```
public List<PapagoInfoVO> getPapagoVOList(PapagoInfoVO tvo) {  
    return pdo.selectPapagoInfoVOList(tvo);  
}
```

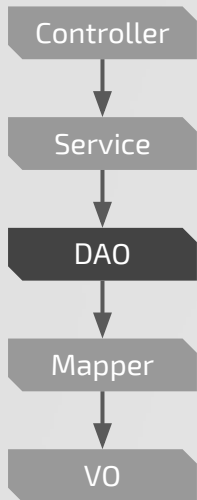
@Override

```
public Map<String, String> deletePapagoInfo(PapagoInfoVO tvo) {  
    Map<String, String> rMap = new HashMap<String, String>();  
    rMap.put("cnt", pdo.deletePapagoInfo(tvo) + "");  
    return rMap;  
}
```

}

PapagoServiceImpl.java





@Repository

```
public class PapagoDAOImpl implements PapagoDAO {
```

@Resource

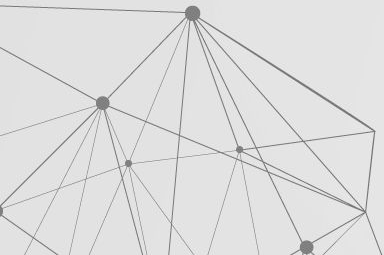
```
private SqlSessionFactory ssf;
```

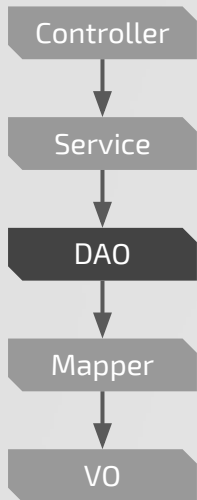
@Override

```
public List<PapagoInfoVO> selectPapagoInfoVOList(PapagoInfoVO tvo) {  
    SqlSession ss = ssf.openSession();  
    try {  
        return ss.selectList("com.jl.papago.dao.PapagoInfoMapper.selectPapagoInfoList", tvo);  
    } catch (Exception e) {  
        e.printStackTrace();  
    } finally {  
        ss.close();  
    }  
    return null;  
}
```

PapagoDAOImpl.java

1 of 2





@Override

```
public int deletePapagoInfo(PapagoInfoVO tvo) {
```

```
    SqlSession ss = ssf.openSession();
```

```
    try {
```

```
        int cnt= ss.delete("com.jl.papago.dao.PapagoInfoMapper.deletePapagoInfo", tvo);
```

```
        ss.commit();
```

```
        if(cnt != tvo.getPNums().length) {
```

```
            ss.rollback();
```

```
            return 0;
```

```
        }
```

```
        return cnt;
```

```
    } catch (Exception e) {
```

```
        e.printStackTrace();
```

```
    } finally {
```

```
        ss.close();
```

```
    }
```

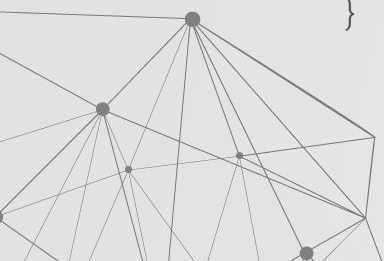
```
    return 0;
```

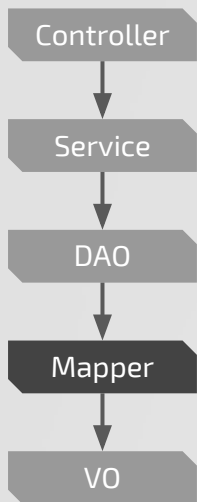
```
}
```

```
}
```

PapagoDAOImpl.java

2 of 2

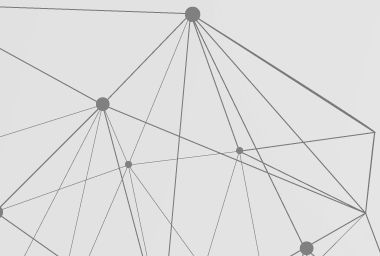


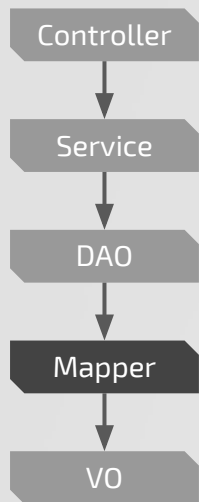


```
<mapper namespace="com.jl.papago.dao.PapagoInfoMapper">
  <select id="selectPapagoInfoList" resultType="tvo">
    select * from papago_info
    <where>
      <if test="search != null">
        <foreach collection="search" item="col" separator="or">
          ${col} like concat('%', #{searchStr}, '%')
        </foreach>
      </if>
    </where>
    <trim>
      <if test='order == null'>
        order by pi_cnt desc
      </if>
      <if test='order != null'>
        order by ${order}
      </if>
    </trim>
  </select>
```

papago-info.xml

1 of 2

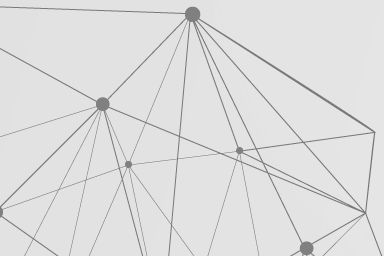


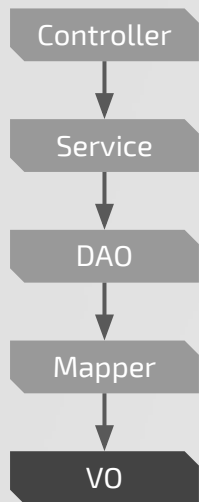


```
<delete id="deletePapagoInfo" parameterType="tvo">
    delete from papago_info
    <where>
        <if test="piNums != null">
            <foreach collection="piNums" item="piNum" separator="or">
                pi_num = #{piNum}
            </foreach>
        </if>
    </where>
</delete>
</mapper>
```

papago-info.xml

2 of 2

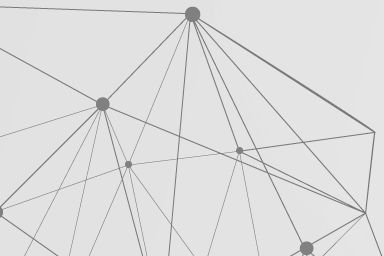




@Data

```
public class PapagoInfoVO {  
    private Integer piNum;  
    private Integer[] piNums;  
    private String piSource;  
    private String piTarget;  
    private String piText;  
    private String piResult;  
    private Integer piCnt;  
    private String[] search;  
    private String searchStr;  
    private String order;  
}
```

PapagoInfoVO.java



Search & Deletion



ERROR

Please select at least one checkbox !

OK



ERROR

Please enter a search keyword !

OK



Deleted!

of Deleted Item(s) : 2

OK

Papago Results !

Source Lang ☒ Target Lang ☒ Source Text ☒ Target Text ☒

SELECT ALL ☒

ge

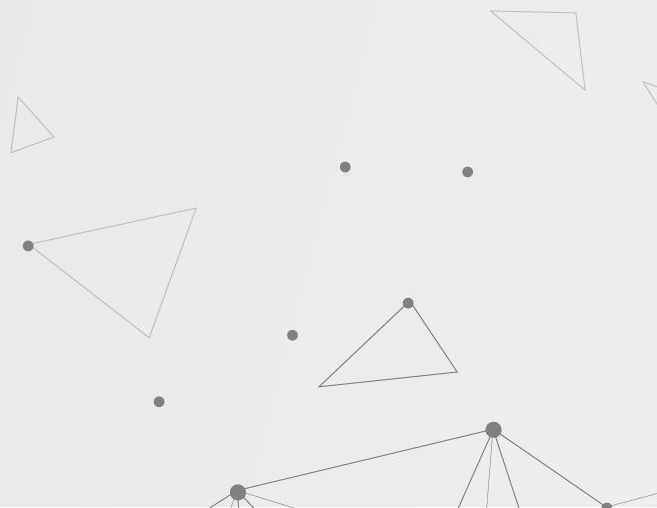
SEARCH

DELETE

Delete	Record # ▼	Source Lang ▼	Target Lang ▼	Source Text ▼	Target Text ▼	Count ▼
<input type="checkbox"/>	149	ko	en	용기	courage	5
<input type="checkbox"/>	877	zh-CN	en	充电器	Charger	3
<input type="checkbox"/>	901	ko	en	귤	tangerine	3
<input type="checkbox"/>	402	ko	en	담당자	The person in charge	3
<input checked="" type="checkbox"/>	261	ko	en	배추	Cabbage	2
<input checked="" type="checkbox"/>	10	ko	en	영어	the English language	2
<input type="checkbox"/>	285	ko	en	도전	Challenge	2

```
<div id="search-and-delete">
  <div id="search-div">
    <div id="option-1">
      <label for="piSource">Source Lang</label>
      <input type="checkbox" id="piSource" class="checkOne" value="pi_source" name="search" />
    </div>
    <div id="option-2">
      <label for="piTarget">Target Lang</label>
      <input type="checkbox" id="piTarget" class="checkOne" value="pi_target" name="search" />
    </div>
    (Code for id : option-3 & option-4)
    <div id="option-all">
      <label>SELECT ALL</label>
      <input type="checkbox" id="checkAll" onclick="selectAll()" />
    </div>
    <div id="search-box-div">
      <input type="text" id="searchStr">
      <button id="search-button">SEARCH</button>
    </div>
    <button id="delete-button" onclick="deleteUsers()">DELETE</button>
  </div>
  <button id="delete-button-large-screen" onclick="deleteUsers()">DELETE</button>
</div>
```

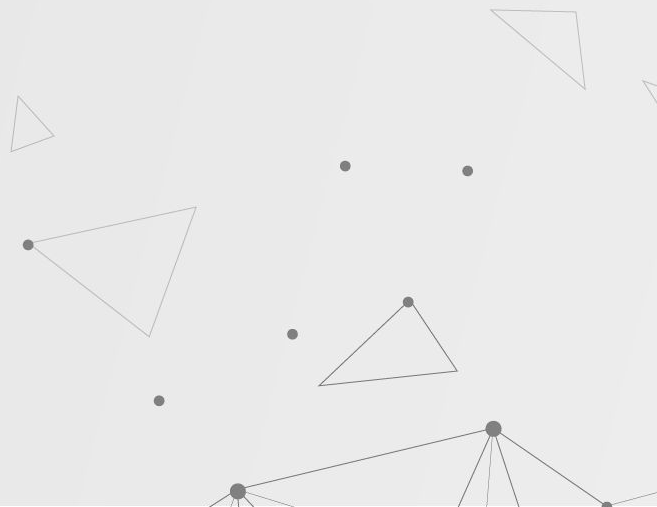
```
<table data-id="all">
  <thead>
    <tr>
      <th>Delete</th>
      <th data-order="pi_num desc">Record # ▼</th>
      <th data-order="pi_source desc">Source Lang ▼</th>
      <th data-order="pi_target desc">Target Lang ▼</th>
      <th data-order="pi_text desc">Source Text ▼</th>
      <th data-order="pi_result desc">Target Text ▼</th>
      <th data-order="pi_cnt desc">Count ▼</th>
    </tr>
  </thead>
  <tbody id="tBody"></tbody>
</table>
```



```
window.onload = function(){  
  var btn = document.querySelector("#search-button");  
  btn.onclick = function(){  
    var checks = document.querySelectorAll("[name=search]:checked");  
    if(checks.length == 0){  
      (code for an alert message)  
    }  
    var searchStr = document.querySelector("#searchStr");  
    if(searchStr.value.trim().length < 1){  
      (code for an alert message)  
    }  
    var search = "";  
    for(var i=0; i<checks.length; i++){  
      Search += search=' + checks[i].value + '&';  
    }  
    search += searchStr=' + searchStr.value;  
    dataLoad(search);  
  }  
}
```

results.js

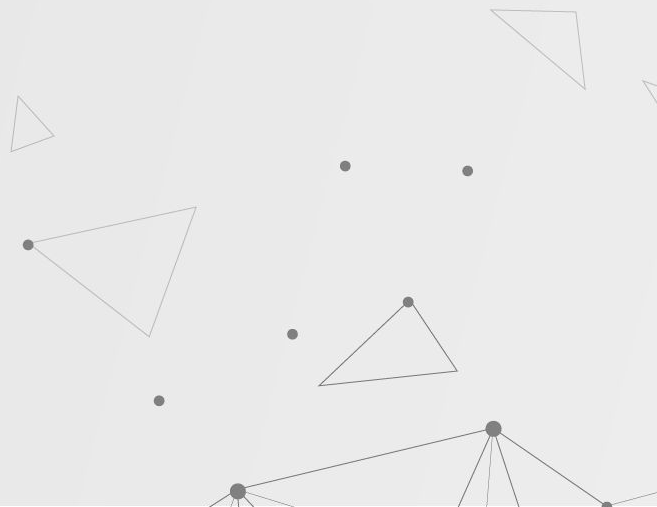
1 of 5




```
var dataOrder = document.querySelectorAll('th[data-order]');

for(var i=0; i<dataOrder.length; i++){
    dataOrder[i].onclick = function(){
        var text = this.innerHTML;
        var symbol = text.substring(text.length -1, text.length);
        var order = this.getAttribute('data-order');

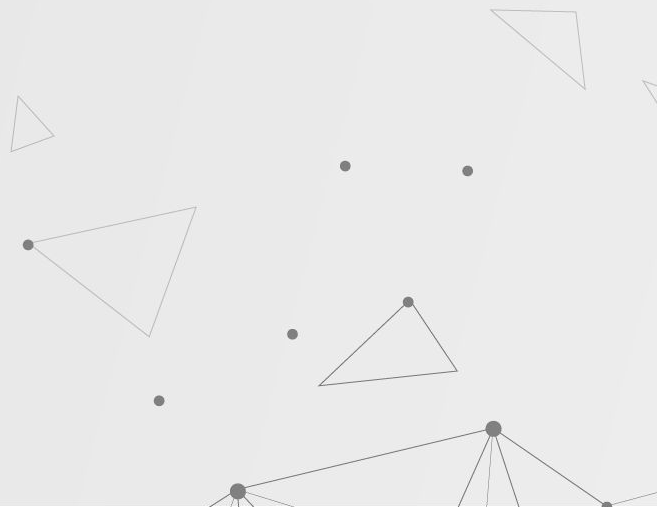
        if(symbol == '▼'){
            this.innerText = text.substring(0, text.length -1) + '▲';
            order = order.substring(0, order.indexOf('desc')) + 'asc';
            this.setAttribute('data-order', order);
        } else if(symbol == '▲') {
            this.innerText = text.substring(0, text.length -1) + '▼';
            order = order.substring(0, order.indexOf('asc')) + 'desc';
            this.setAttribute('data-order', order);
        }
        var param = 'order='+ this.getAttribute('data-order');
        dataLoad(param);
    }
}
dataLoad();
}
```



```
function dataLoad(data){  
    var xhr = new XMLHttpRequest();  
    xhr.open('GET', '/papago?' + data);  
    xhr.setRequestHeader('Content-Type', 'application/json');  
    xhr.onreadystatechange = function(){  
        if(xhr.readyState == 4 && xhr.status == 200){  
            var list3 = JSON.parse(xhr.responseText);  
            var tBody = document.getElementById('tbody');  
            var html = "";  
            for(var item of list3){  
                html += '<tr>';  
                html += '<td><input type="checkbox" name="remove" value="" + item.piNum + "></td>';  
                html += '<td>' + item.piNum + '</td>';  
                html += '<td>' + item.piSource + '</td>';  
                html += '<td>' + item.piTarget + '</td>';  
                html += '<td>' + item.piText + '</td>';  
                html += '<td>' + item.piResult + '</td>';  
                html += '<td>' + item.piCnt + '</td>';  
                html += '</tr>';  
            }  
            tBody.innerHTML = html;  
        }  
    }  
    xhr.send();  
}
```

```
function selectAll() {  
    var selectAll = document.getElementById('checkAll').checked;  
    var checkOptions = document.getElementsByClassName('checkOne');  
    var intLength = checkOptions.length;  
    for(var i=0; i<intLength; i++){  
        var checkOption = checkOptions[i];  
        checkOption.checked = selectAll;  
    }  
}
```

```
function deleteUsers() {  
    var checks = document.querySelectorAll('input[name=remove]:checked');  
    if(checks.length<1){  
        Swal.fire({  
            position: 'top',  
            icon: 'error',  
            title: 'ERROR',  
            text: 'Please select at least one item to delete !'  
        })  
        return;  
    }  
}
```

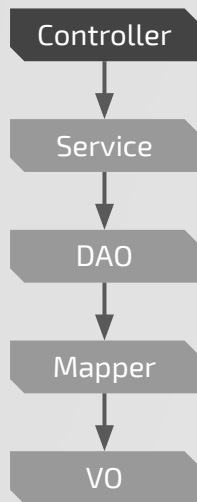


```
var piNums = "";
for(var i=0; i<checks.length; i++){
    piNums += 'piNums=' + [checks[i].value] + '&';
}
var xhr2 = new XMLHttpRequest();
xhr2.open('DELETE', '/papago?' + piNums);
xhr2.setRequestHeader('Content-Type', 'application/json');
xhr2.onreadystatechange = function(){
    if(xhr2.readyState == 4 && xhr2.status == 200){
        var res = JSON.parse(xhr2.responseText);
        if(res.cnt != 0){
            Swal.fire({
                position: 'top',
                icon: 'success',
                title: 'Deleted!',
                text: '# of Deleted Item(s) : ' + res.cnt
            }).then(() => {
                location.href = '/views/papago/index?page=results';
            })
        }
    }
}
xhr2.send(piNums);
}
```

04

SORTING

Sorting translated texts *by* user & date as well as *by* an ascending/descending order



@RestController

```
public class PapagoStatsController {
```

PapagoStatsController.java

@Resource

```
private PapagoStatsService psService;
```

@GetMapping("/papago/stats")

```
public List<PapagoStatsVO> getPapagoStatsList(PapagoStatsVO ps){  
    return psService.selectPapagoStatsList(ps);  
}
```

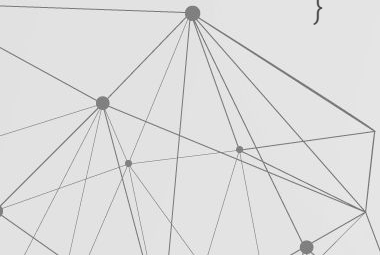
@GetMapping("/papago/stats/date")

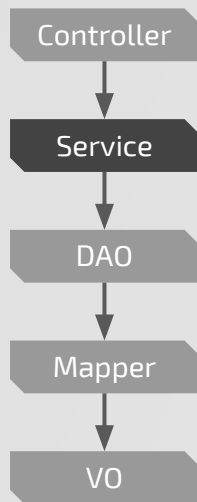
```
public List<PapagoStatsVO> selectPapagoStatsListByDate(PapagoStatsVO ps){  
    return psService.selectPapagoStatsListByDate(ps);  
}
```

@GetMapping("/papago/stats/user")

```
public List<PapagoStatsVO> selectPapagoStatsListByUser(PapagoStatsVO ps){  
    return psService.selectPapagoStatsListByUser(ps);  
}
```

```
}
```





`@Service("papagoService")`

```
public class PapagoStatsServiceImpl implements PapagoStatsService {
```

PapagoServiceImpl.java

`@Resource`

```
private PapagoStatsDAO psdao;
```

`@Override`

```
public List<PapagoStatsVO> selectPapagoStatsList(PapagoStatsVO ps) {  
    return psdao.selectPapagoStatsList(ps);  
}
```

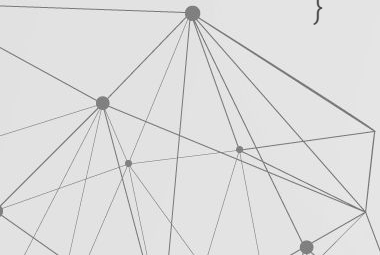
`@Override`

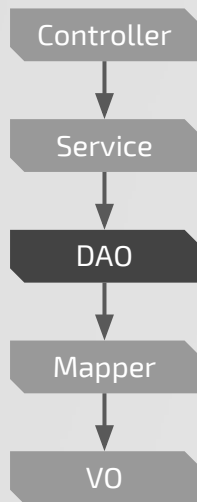
```
public List<PapagoStatsVO> selectPapagoStatsListByDate(PapagoStatsVO ps) {  
    return psdao.selectPapagoStatsListByDate(ps);  
}
```

`@Override`

```
public List<PapagoStatsVO> selectPapagoStatsListByUser(PapagoStatsVO ps) {  
    return psdao.selectPapagoStatsListByUser(ps);  
}
```

```
}
```





@Repository

```
public class PapagoStatsDAOImpl implements PapagoStatsDAO {
```

@Resource

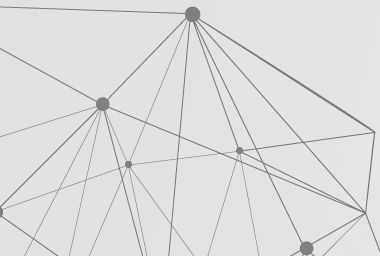
```
private SqlSessionFactory ssf;
```

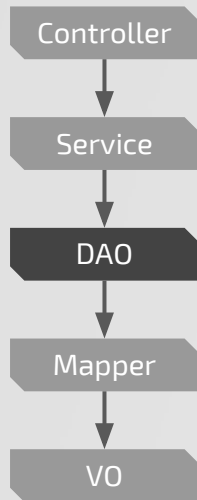
@Override

```
public List<PapagoStatsVO> selectPapagoStatsList(PapagoStatsVO ps) {  
    SqlSession ss = ssf.openSession();  
    try {  
        return ss.selectList("com.jl.papago.dao.PapagoStatsMapper.selectPapagoStatsList", ps);  
    } catch (Exception e) {  
        log.error(e.getMessage());  
    } finally {  
        ss.close();  
    }  
    return null;  
}
```

PapagoStatsDAOImpl.java

1 of 2



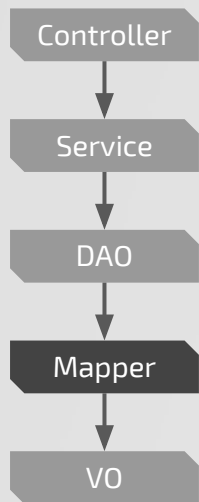


@Override

```
public List<PapagoStatsVO> selectPapagoStatsListByDate(PapagoStatsVO ps) {  
    SqlSession ss = ssf.openSession();  
    try {  
        return ss.selectList("com.jl.papago.dao.PapagoStatsMapper.selectPapagoStatsListByDate", ps);  
    } catch (Exception e) {  
        log.error(e.getMessage());  
    } finally {  
        ss.close();  
    }  
    return null;  
}
```

@Override

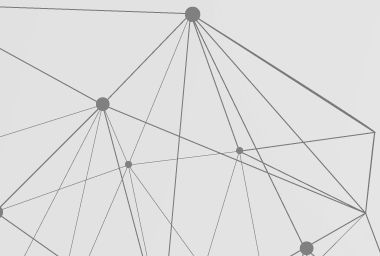
```
public List<PapagoStatsVO> selectPapagoStatsListByUser(PapagoStatsVO ps) {  
    SqlSession ss = ssf.openSession();  
    try {  
        return ss.selectList("com.jl.papago.dao.PapagoStatsMapper.selectPapagoStatsListByUser", ps);  
    }  
    (same as above)  
}
```

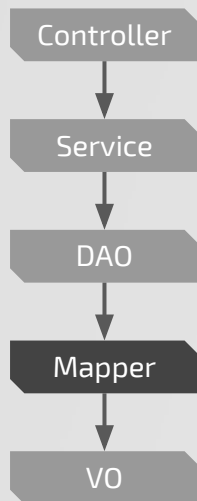


```
<mapper namespace="com.jl.papago.dao.PapagoStatsMapper">
  <select id="selectPapagoStatsList" resultType="ps">
    select ps.*, ui.ui_id
    from papago_stats ps
    inner join user_info ui
    on ps.ui_num = ui.ui_num
    <trim>
      <if test='order == null'>
        order by ps_num desc
      </if>
      <if test='order != null'>
        order by ${order}
      </if>
    </trim>
  </select>
```

papago-stats.xml

1 of 2



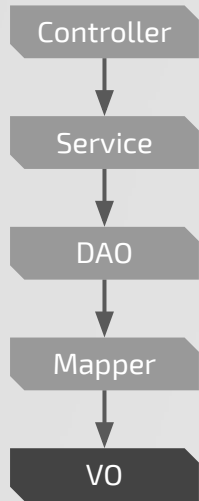


```
<select id="selectPapagoStatsListByDate" resultType="ps">
    select count(1) as cnt, ps.credat
    from papago_stats ps
    group by ps.credat
    <trim>
        <if test='order == null'>
            order by cnt desc
        </if>
        <if test='order != null'>
            order by ${order}
        </if>
    </trim>
</select>
```

```
<select id="selectPapagoStatsListByUser" resultType="ps">
    select ps.ui_num, ui.ui_id, count(1) as cnt
    from papago_stats ps
    inner join user_info ui on ps.ui_num = ui.ui_num
    group by ps.ui_num, ui.ui_id
    <trim>
        (same as above)
```

```
</select>
```

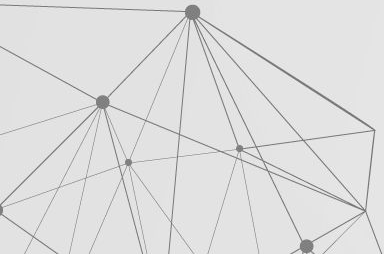
```
</mapper>
```



@Data

```
public class PapagoInfoVO {  
    private Integer psNum;  
    private Integer uiNum;  
    private Integer piNum;  
    private String uild;  
    private String credat;  
    private Integer cnt;  
    private String order;  
}
```

PapagoInfoVO.java



Stats (*All, User, Date*)

Stats list

By Date

By User

All

Request User # ▼	Request User ID ▼	By # of Searches ▼
26	summer	788
28	oneus	203
23	leedo	183
29	hyejung	111
30	lucky	32
22	jaylee	23
27	jenny	20
31	happy	7
32	jen-lim	1

By Date

By User

All

Record # ▼	Request User # ▼	Request User ID ▼	Translation # ▼	Date ▼
1688	26	summer	149	20200202
1687	26	summer	945	20200202
1686	26	summer	945	20200202
1685	26	summer	945	20200202
1684	26	summer	944	20200202
1683	26	summer	943	20200202
1682	26	summer	943	20200202
1681	28	oneus	942	20200201
1680	28	oneus	941	20200201
1679	28	oneus	216	20200201

By Date

By User

All

By Date ▼	By # of Searches ▼
20191231	258
20191225	231
20200102	219

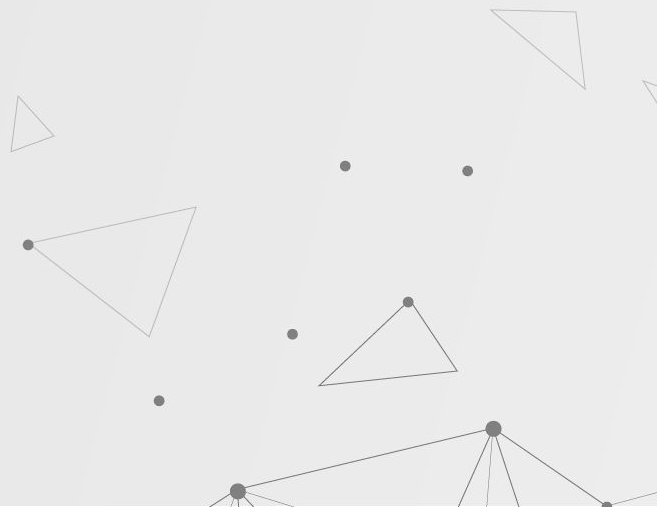
By Date

By User

All

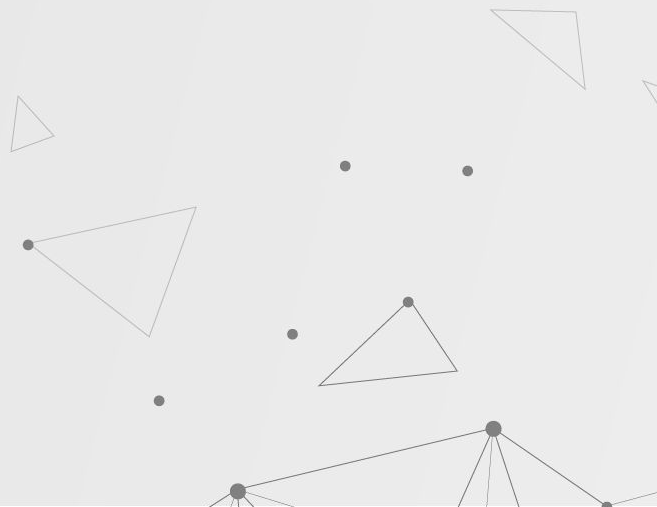
By Date ▲	By # of Searches ▼
20191223	17
20191224	24
20191225	231

```
<h1 class="animated lightSpeedIn">Stats list</h1>
<button type="button" data-target="credat">By Date</button>
<button type="button" data-target="user">By User</button>
<button type="button" data-target="all">All</button><br><br>
<table data-id="all" id="all-div">
  <thead>
    <tr>
      <th data-order="ps.ps_num desc">Record # ▼</th>
      <th data-order="ui.ui_num desc">Request User # ▼</th>
      <th data-order="ps.ui_num desc">Request User ID ▼</th>
      <th data-order="pi_num desc">Translation # ▼</th>
      <th data-order="ps.credat desc">Date ▼</th>
    </tr>
  </thead>
  <tbody id="tBody"></tbody>
</table>
```



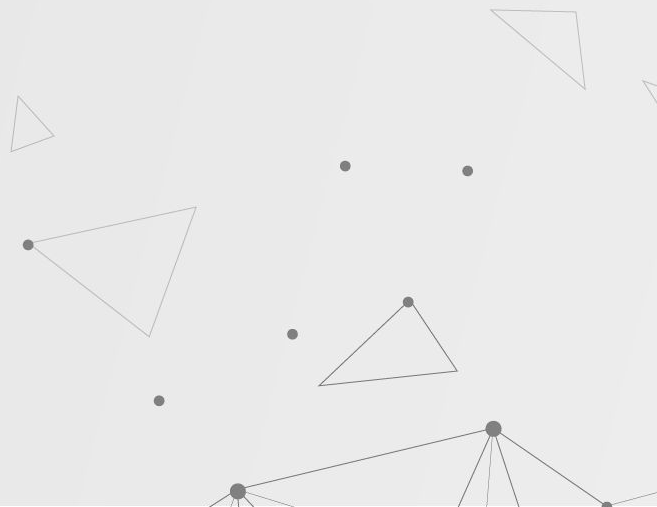
```
<table data-id="credat" style="display:none;">
  <thead>
    <tr>
      <th data-order="ps.credat desc">By Date ▼</th>
      <th data-order="cnt desc">By # of Searches ▼</th>
    </tr>
  </thead>
  <tbody id="tBody-credat"></tbody>
</table>
```

```
<table data-id="user" style="display:none;">
  <thead>
    <tr>
      <th data-order="ps.ui_num desc">Request User # ▼</th>
      <th data-order="ui.ui_id desc">Request User ID ▼</th>
      <th data-order="cnt desc">By # of Searches ▼</th>
    </tr>
  </thead>
  <tbody id="tBody-user"></tbody>
</table>
```



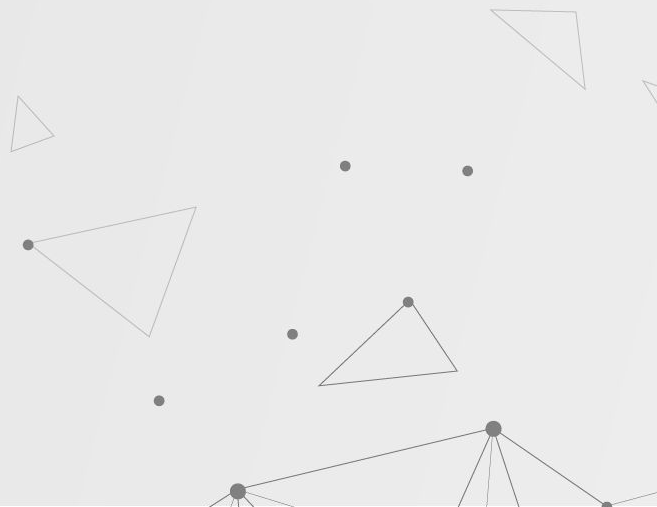
```
$(document).ready(function(){
    $('button[data-target]').on('click', function(){
        $('table[data-id]').css('display', 'none');
        $('table[data-id=' + this.getAttribute('data-target') + ']').css('display', '');

        if(this.getAttribute('data-target') == 'credat'){
            dataLoadByDate();
        } else if(this.getAttribute('data-target') == 'user'){
            dataLoadByUser();
        }
    })
    $('th[data-order]').on('click', function(){
        var text = this.innerHTML;
        var symbol = text.substring(text.length - 1, text.length);
        var order = this.getAttribute('data-order');
        if(symbol == '▼'){
            this.innerHTML = text.substring(0, text.length-1) + '▲';
            order = order.substring(0, order.indexOf('desc')) + 'asc';
            this.setAttribute('data-order', order);
        } else if(symbol == '▲') {
            this.innerHTML = text.substring(0, text.length-1) + '▼';
            order = order.substring(0, order.indexOf('desc')) + 'desc';
            this.setAttribute('data-order', order);
        }
    })
})
```

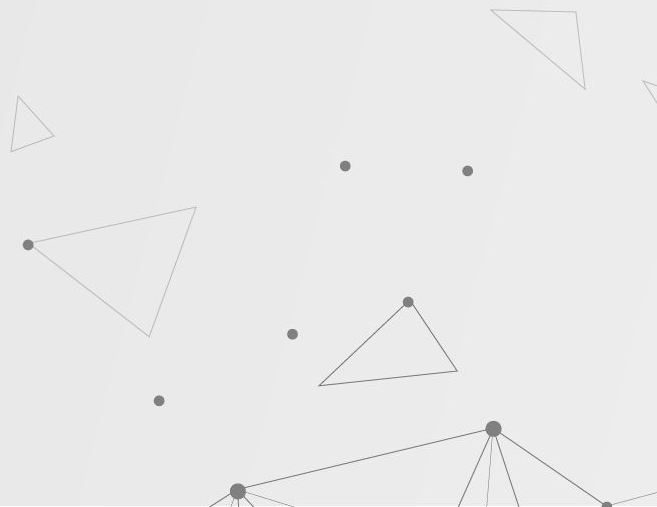



```
var param = 'order='+ this.getAttribute('data-order');

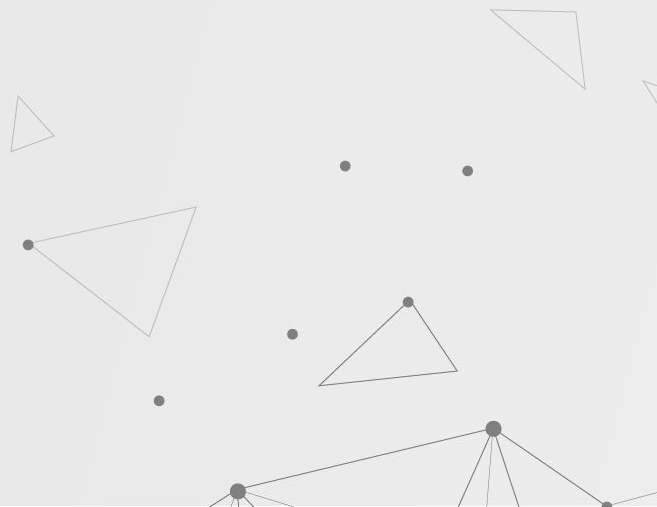
if(this.parentNode.parentNode.parentNode.getAttribute('data-id') == 'credat'){
    dataLoadByDate(param);
} else if(this.parentNode.parentNode.parentNode.getAttribute('data-id') == 'user'){
    dataLoadByUser(param);
} else {
    dataLoad(param);
}
})
dataLoad();
})
```



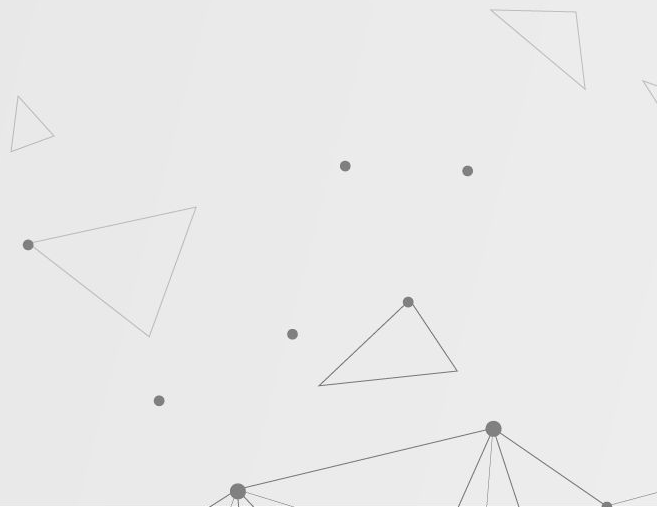
```
function dataLoad(data){  
  $.ajax({  
    url: "/papago/stats?" + data,  
    method: 'GET',  
    data: data,  
    success: function(res){  
      var html = "";  
      for(var item of res){  
        html += '<tr>';  
        html += '<td>' + item.psNum + '</td>';  
        html += '<td>' + item.uiNum + '</td>';  
        html += '<td>' + item.uild + '</td>';  
        html += '<td>' + item.piNum + '</td>';  
        html += '<td>' + item.credat + '</td>';  
        html += '</tr>';  
      }  
      $('#tbody').html(html);  
    },  
    error: function(res){  
      console.log(res);  
    }  
  })  
}
```



```
function dataLoadByDate(data){  
  $.ajax({  
    url: "/papago/stats/date?" + data,  
    method: 'GET',  
    data: data,  
    success: function(res){  
      var html = "";  
      for(var item of res){  
        html += '<tr>';  
        html += '<td>' + item.credat + '</td>';  
        html += '<td>' + item.cnt + '</td>';  
        html += '</tr>';  
      }  
      $('#tbody-credat').html(html);  
    },  
    error: function(res){  
      console.log(res);  
    }  
  })  
}
```



```
function dataLoadByUser(data){
    $.ajax({
        url: "/papago/stats/user?" + data,
        method: 'GET',
        data: data,
        success: function(res){
            var html = "";
            for(var item of res){
                html += '<tr>';
                html += '<td>' + item.uiNum + '</td>';
                html += '<td>' + item.uid + '</td>';
                html += '<td>' + item.cnt + '</td>';
                html += '</tr>';
            }
            $('#tbody-user').html(html);
        },
        error: function(res){
            console.log(res);
        }
    })
}
```



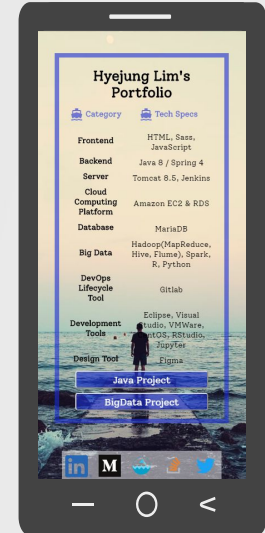
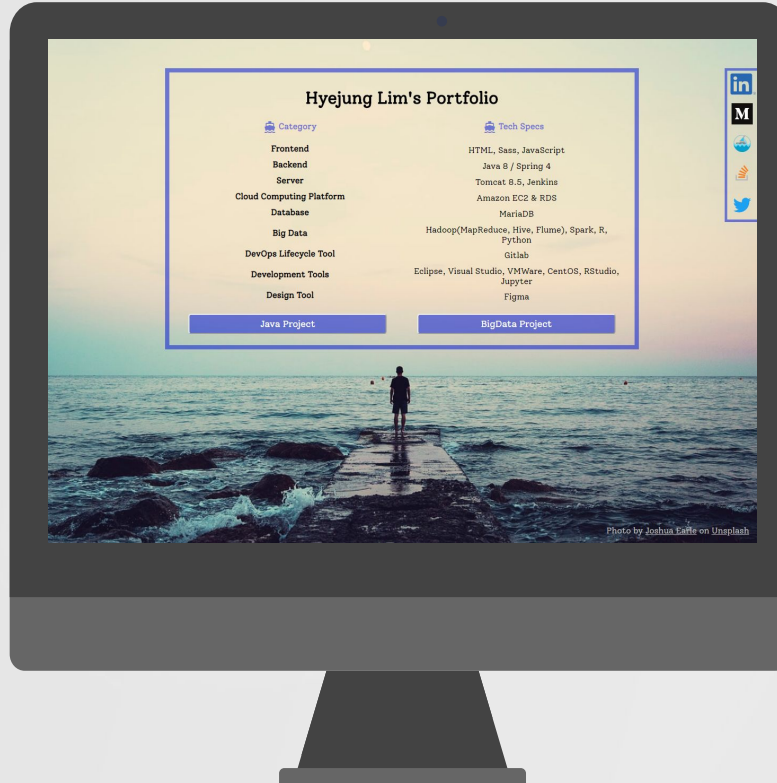
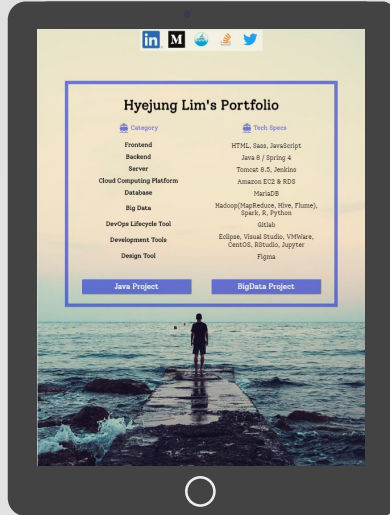
The background features a complex network of thin grey lines connecting various-sized dark grey circular nodes. These nodes are scattered across the slide, with a higher concentration on the right side. A vertical grey line is positioned to the left of the main text.

05

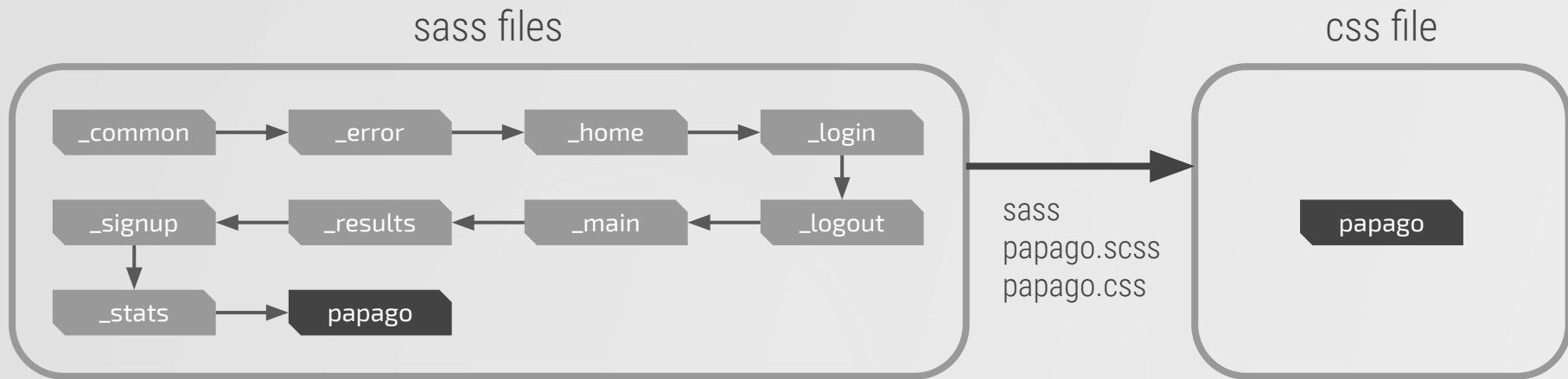
DESIGN

Responsive web design on Chrome

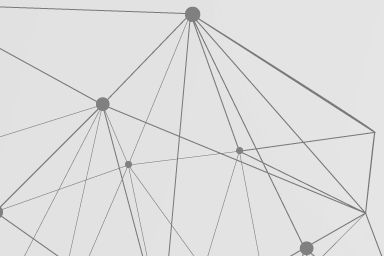
RESPONSIVE DESIGN



CSS Files



Files are connected with `@import "(file name)";`



Media Queries

/* Extra small devices (phones, 600px and down) */

@media only screen and (max-width: 600px) {

/* Small devices (portrait tablets and large phones, 600px and up) */

@media only screen and (min-width: 600px) and (max-width: 768px) and (orientation: landscape)

/* small-sized window screens */

@media only screen and (min-width: 600px) and (max-width: 768px) and (orientation: portrait)

/* Medium devices (landscape tablets, 768px and up) */

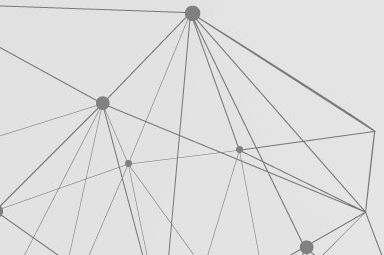
@media only screen and (min-width: 768px) and (max-width: 992px) and (orientation: landscape)

/* medium-sized window screens */

@media only screen and (min-width: 768px) and (max-width: 992px) and (orientation: portrait)

/* Large devices (laptops/desktops, 992px and up) */

@media only screen and (min-width: 992px)





06

DEPLOYING A WEBSITE

Deploying a website on Amazon EC2 & RDS (*ft. Jenkins*)

STEPS (1 of 7)

1. AWS EC2

- ◆ **[AWS]** EC2 Dashboard > Launch Instance > Red Hat Enterprise Linux 8 (Free tier) > Download Key Pair
- ◆ **[AWS]** Elastic IPs > Allocate Elastic IP address > Associate
- ◆ **[AWS]** Security Groups > Custom TCP > Port Range :8080
- ◆ **[Putty]** Putty Key Generator > Load > Save private key
- ◆ **[AWS]** Instances > Elastic IPs #
- ◆ **[Putty]** Session > Host Name (*Elastic IP goes here*) + Port 22
- ◆ **[Putty]** Connection > Auth > Browse > (*Private key file goes here*)

For more details, please visit <http://bit.ly/aws-jenkins>.

STEPS (2 of 7)

2. Authentication, Java, Tomcat, Maven, Git

- ◆ [Putty] Session > Host Name (*Elastic IP goes here*) + Port 22
- ◆ [Putty] Connection > Auth > Browse > (*Private key file goes here*)
- ◆ Login to Putty
- ◆ [Putty] \$ sudo su
- ◆ [Putty] # yum install java-1.8.0-openjdk-devel -y
- ◆ [Putty] # cd home > dd if=/dev/zero of=/newswap bs=1024 count=2000000
- ◆ [Putty] # cd 0600 /newswap > mkswap /newswap > swapon /newswap > swapon
- ◆ [Putty] # wget (*tomcat tar.gz : copy link address*) > tar -xvf (*tomcat tar gz file*)
- ◆ [Putty] # yum install maven -y > yum install git -y
- ◆ [Putty] # mkdir gitrepo > cd gitrepo > git clone (*gitlab address*) > mvn clean install

For more details, please visit <http://bit.ly/aws-jenkins>.

STEPS (3 of 7)

3. AWS RDS

- ◆ [AWS] RDS > Create database > Standard Create > MariaDB > Free tier > (Uncheck) Enable storage autoscaling > Publicly accessible : Yes > VPC security group : Create new > Database port :3003

4. MariaDB

- ◆ [MariaDB] Download > Enable networking - TCP port :3003
- ◆ [HeidiSQL] New > Hostname/IP : 127.0.0.1 > Port :3003 > Open > Create tables
- ◆ [AWS] RDS > Databases > (Select DB) > Copy Endpoint
- ◆ [HeidiSQL] New > Hostname/IP : (Paste Endpoint) > Port :3003 > Open > File > Session manager
- ◆ [HeidiSQL] Right click a database in Hostname/IP : 127.0.0.1 > Export database as SQL
- ◆ [HeidiSQL] Data : Insert > Filename : Select the exported file
- ◆ [HeidiSQL] File > Load SQL file > Run queries

For more details, please visit <http://bit.ly/aws-jenkins>.

STEPS (4 of 7)

- ◆ **[AWS]** Network & Security > Security Groups > Copy GroupID for the database > Inbound > Edit
- ◆ **[AWS]** Add Rule > Port Range :3003 > Source : *(Paste the GroupID)* > Copy Endpoint
- ◆ **[AWS]** RDS > Parameter groups > Create parameter group > lower_case_table_names > Values : 1
- ◆ **[AWS]** RDS > Parameter groups > Create parameter group > character_set_client > Values : utf8
- ◆ **[AWS]** RDS > Databases > *(Select the DB)* > DB parameter group > *(Select the new one)*
- ◆ **[AWS]** RDS > Databases > Actions > Reboot

5. Tomcat Manager

- ◆ **[Putty]** # cd gitrepo/jl-papago/target > mv ./transper.war /home/tomcat8/webapps/init.war
- ◆ **[Putty]** # cd .. > cd conf/ > vi server.xml > Line 69 : change port to "80"
 - > Line 163 : add "<Context docBase='init' path='/' reloadable='true'>"

For more details, please visit <http://bit.ly/aws-jenkins>.

STEPS (5 of 7)

- ◆ **[Putty]** # vi tomcat-users.xml > Line 44 : add "<role rolename='manager-gui'/>"
 - > Line 45 : add "<role rolename='manager-script'/>" > Line 46 : add "<role rolename='manager-status'/>"
 - > Line 47 : add "<user username='admin' password='(Enter Password)'
roles='manager-gui, manager-script, manager-status'/>"
- ◆ **[Putty]** # cd /home/tomcat8/webapps/manager/META-INF > vi context.xml

6. Jenkins, Gitlab

- ◆ **[Putty]** # rpm --import https://jenkins-ci.org/redhat/jenkins-ci.org.key
- ◆ **[Putty]** # wget -O /etc/yum.repos.d/jenkins.repo http://pkg.jenkins-ci.org/redhat/jenkins.repo
- ◆ **[Putty]** # yum install jenkins -y > systemctl start jenkins
- ◆ **[AWS]** EC2 > Network & Security > Security Groups > *(Click the group name for connection)* > Inbound > Edit
 - > Type : Custom TCP > Port Range : 8080 > Description : jenkins

For more details, please visit <http://bit.ly/aws-jenkins>.

STEPS (6 of 7)

- ◆ **[Jenkins]** (*Elastic IP address*):8080/login
- ◆ **[Putty]** # cd /home/tomcat8/bin
- ◆ **[Putty]** # vi var/lib/jenkins/secrets/initialAdminPassowrd > (*Copy the password*)
- ◆ **[Jenkins]** Paste the password > Install suggested plugins
- ◆ **[Jenkins]** Manage Jenkins > Global Tool Configuration > [Git] Path to Git executable : /bin/git
 - > [Maven] Name : Maven 3.5.4 > MAVEN_HOME : /usr/share/maven > *Check* Install automatically
- ◆ **[Jenkins]** Manage Jenkins > Manage Plugins > *Check* Maven Integration & Deploy to container
- ◆ **[Jenkins]** (*Restart Jenkins*) > Create new jobs > Free project
 - > Source Code Management > Git > Repositories - URL : (*Enter GitLab URL*)
 - > Source Code Management > Git > Credentials : (*Enter GitLab ID & Password*)
 - > Build > Add build step > Invoke top-level Maven targets > Goals : clean install

For more details, please visit <http://bit.ly/aws-jenkins>.

STEPS (7 of 7)

- Post-build Actions ➤ Add Post-build action ➤ Deploy war/ear to a container
- WAR/EAR files : `**/*.war` ➤ Context path : `/` ➤ Container : [\(Add Tomcat Credentials & URL\)](#)
- ◆ **[Jenkins]** Manage Jenkins ➤ Manage Plugins ➤ [Check](#) GitLab & GitLab Plugin
- ◆ **[Jenkins]** [\(Click Project\)](#) ➤ Configure ➤ Build Triggers ➤ [Check](#) Build when a change is pushed to GitLab
 - [\(Copy the GitLab webhook URL\)](#) ➤ Advanced ➤ [Click](#) Generate "Secret Token" ➤ [\(Copy the token\)](#)
- ◆ **[GitLab]** [\(Click Project\)](#) ➤ Settings ➤ Integrations ➤ Integrations ➤ Paste URL & Secret Token from Jenkins
 - [Click](#) "Add webhook" ➤ Test ➤ Push events
- ◆ **[Putty]** `# cd /home/tomcat8/conf/ ➤ vi server.xml ➤ Comment everything in Line 163`
- ◆ **[Jenkins]** [\(Click Project\)](#) ➤ Configure ➤ Build Environment ➤ [Check](#) GitHub hook trigger for GITScm polling
- ◆ **[Putty]** `# cd /home/tomcat8/bin ➤ ./startup.sh`

For more details, please visit <http://bit.ly/aws-jenkins>.

THANKS

Please visit <http://bit.ly/hyejung-jen-lim> to check out this project.

Hyejung(*Jen*) Lim
hyejung_jen_lim@naver.com
010-4494-5818

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**.

Please keep this slide for attribution.