전자정부 표준프레임워크

실행환경(화면처리) 실습교재

eGovFrame

Contents

실행환경(화면처리) 실습교재

Lab 301 – mvc

eGovFrame

실습 개요

- 실습을 통해 MVC에 대하여 살펴본다.
- 실습 순서

- Hello world예제

1. Controller 일부 작성
2. JSP 일부 작성

- 로그인 예제

1. XML 설정
2. LoginController 작성
3. LoginCommand 작성

- 세션 및 국제화 예제

1. XML 설정
2. LoginController에 @SessionAttribute, 메소드 추가

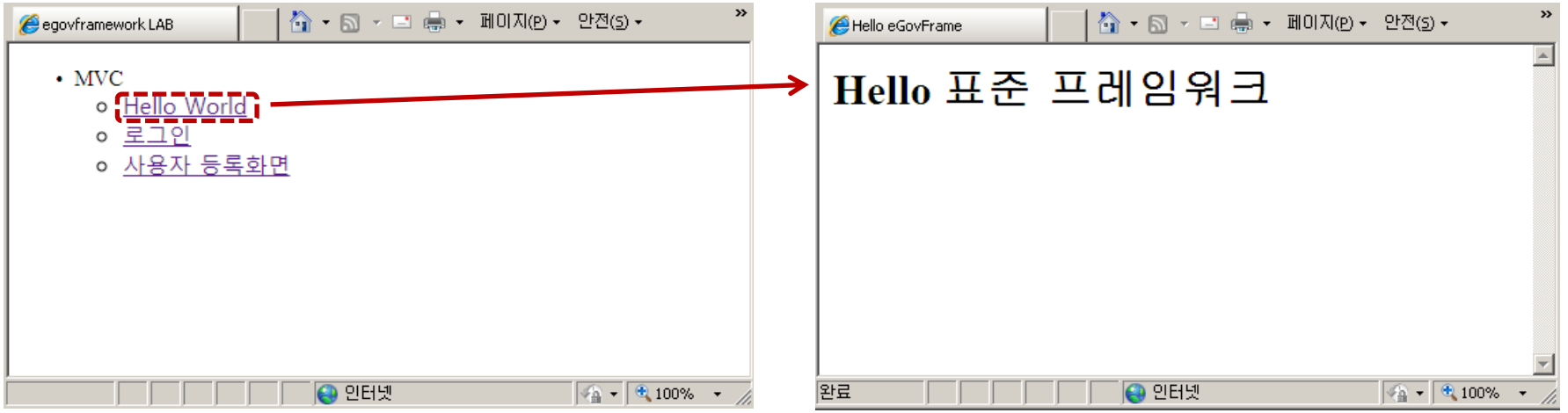
실행환경(화면처리) 실습교재

LAB 301-mvc 실습(1)

eGovFrame

WHAT TO DO

'Hello World' 예제 실행결과 확인



- 프로젝트 선택 마우스 우클릭 > Run As > Run On Server 실행
- 예제 실행 결과 확인 (http://127.0.0.1:8080/lab301-mvc/)

Step 1-1-1. ViewResolver - View를 처리할 해결사를 설정하자

```
<bean  class="org.springframework.web.servlet.view.InternalResourceViewResolver"
       p:prefix="/WEB-INF/jsp/" p:suffix=".jsp" />
```

Step 1-1-2. @RequestMapping – 요청 URL과 View 연결하기

```
@RequestMapping(value = "/hello.do")
   public String helloworld() {
      return getViewName();
   }
```

Step 1-1-3. View 만들기 - helloworld.jsp 만들기 (위치 : src₩main₩webapp₩WEB-INF₩jsp₩hello)

```
<%@ page contentType="text/html; charset=utf-8" pageEncoding="utf-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form"%>
<%@ taglib prefix="spring" uri="http://www.springframework.org/tags"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Hello eGovFrame</title>
</head>
<body>
<h1>Hello 표준 프레임워크  </h1>
</body>
</html>
```

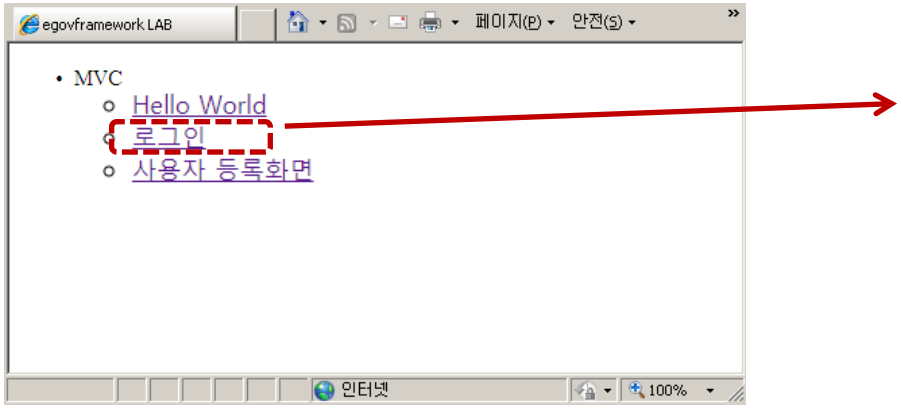
실행환경(화면처리) 실습교재

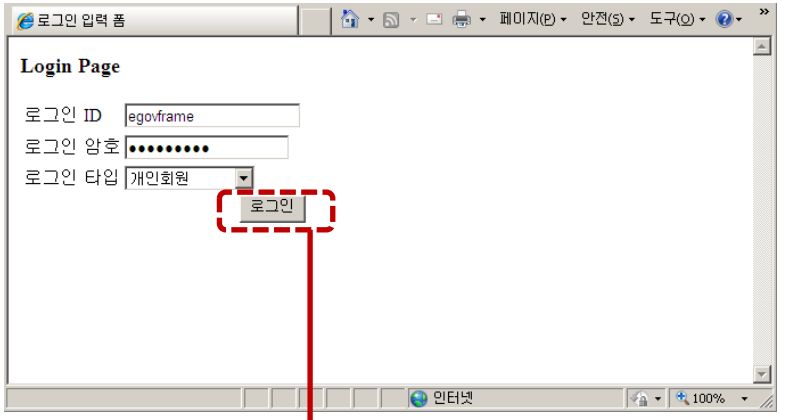
LAB 301-mvc 실습(2)

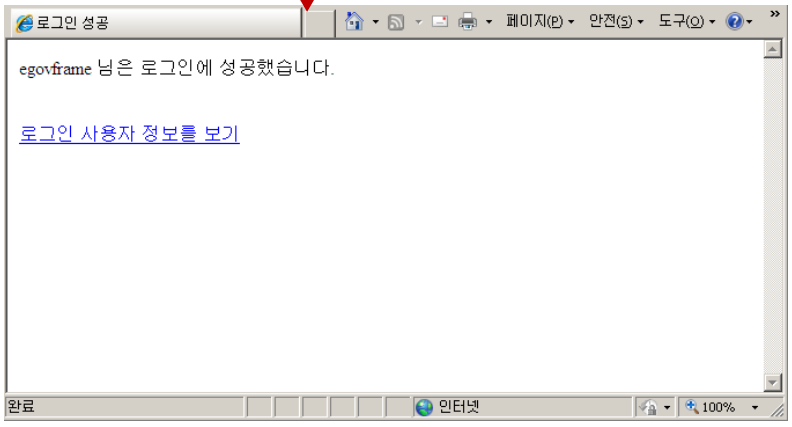
eGovFrame

WHAT TO DO

'로그인' 예제 실행결과 확인







프로젝트 선택 마우스 우클릭 > Run As >

Run On Server 실행

예제 실행 결과 확인 (http://127.0.0.1:8080/lab301-mvc/)

Step 1-2-1. SpringMessage - messageSource 활성화 설정 (in context-servlet.xml)

```
<bean id="messageSource"
      class="org.springframework.context.support.ResourceBundleMessageSource">
      <property name="basenames">
         <list>
            <value>messages.message-common</value>
         </list>
      </property>
   </bean>
```

Step 1-2-2. SpringMessage - <spring:message/> 사용

```
<td>
    <label for="id">
        <spring:message code="login.form.id" />
    </label>
</td>
<td>
    <form:input id="id" path="id" />
</td>
<td>
    <form:errors path="id" />
</td>
```

Step 1-2-3. LoginCommand.java 완성하기

```
private String id;
private String password;
private String loginType;

public String getId() {
return id;
}

public void setId(String id) {
this.id = id;
}

public String getPassword() {
return password;
}

public void setPassword(String password) {
this.password = password;
}

public String getLoginType() {
return loginType;
}

public void setLoginType(String loginType) {
this.loginType = loginType;
}
```

Step 1-2-4. @RequestMapping - method별 mapping 전략

```
@RequestMapping(value = "/loginProcess1.do", method = RequestMethod.GET)
public String loginFormSetUp() {
    return getFormView();
}

@RequestMapping(value = "/loginProcess1.do", method = RequestMethod.POST)
public String loginProcess(@ModelAttribute("login") LoginCommand loginCommand) {

    return getSuccessView();
}
```

Step 1-2-5. @ModelAttribute - 모델의 초기화

```
@ModelAttribute("loginTypes")
protected List<LoginType> referenceData() throws Exception {
    List<LoginType> loginTypes = new ArrayList<LoginType>();
    loginTypes.add(new LoginType("A", "개인회원"));
    loginTypes.add(new LoginType("B", "기업회원"));
    loginTypes.add(new LoginType("C", "관리자"));
    return loginTypes;
}

@ModelAttribute("login")
protected Object referenceData4login() throws Exception {
    return new LoginCommand();
}
```

실행환경(화면처리) 실습교재

LAB 301-mvc 실습(3)

eGovFrame

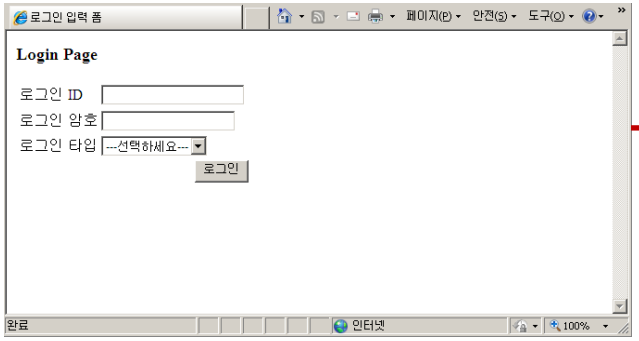
WHAT TO DO

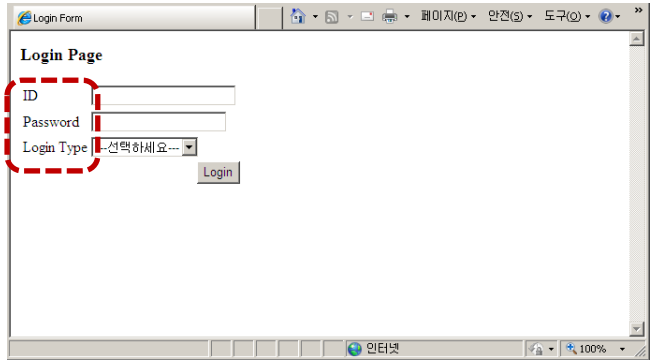
국제화 적용 결과 확인

- 로그인 페이지 locale 변경(en)

  (http://127.0.0.1:8080/lab301-mvc-tutor/loginProcess1.do?lang=en)

- 예제 실행 결과 확인



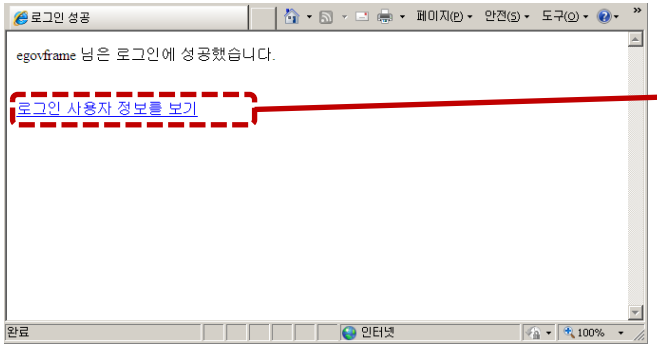


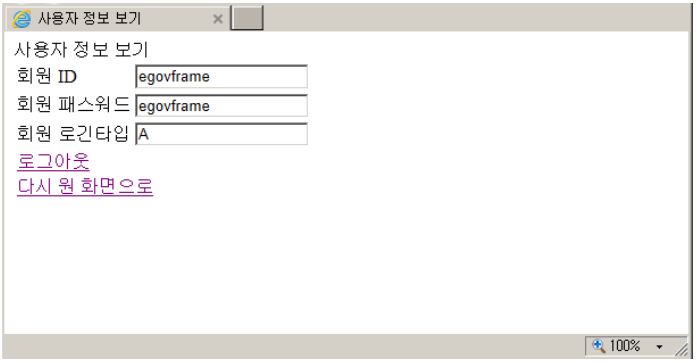
'사용자 정보보기' 예제 실행결과 확인

- 예제 실행 결과 확인
- get 방식 호출임에도

  Login 객체들이

  보이는 이유는?





Step 1-3-1. Internalization - 국제화 관련 bean 설정

* HandlerMapping 설정과 함께 보면 좋습니다.

```
<bean id="localeChangeInterceptor"
      class="org.springframework.web.servlet.i18n.LocaleChangeInterceptor"
      p:paramName="lang" />

<bean id="localeResolver" class="org.springframework.web.servlet.i18n.SessionLocaleResolver" />

<bean class="org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerMapping">
   <property name="interceptors">
            <list>
                <ref bean="localeChangeInterceptor"/>
            </list>
        </property>
   </bean>

<bean class="org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter" />
```

Step 1-3-2. LoginController.java 에 @SessionAttributes 설정 하기

```
@Controller
@SessionAttributes("login")
public class LoginController
```

Step 1-3-3. LoginController.java 에 @SessionAttributes 설정 하기

```
  @RequestMapping(value = "/memberInfo.do")
   public ModelAndView memberInfo(HttpSession httpSession) {
       ModelAndView mav = new ModelAndView("login/memberInfo");

//         if (httpSession.getAttribute("login") != null) {
//             mav.addObject("login", httpSession.getAttribute("login"));
//         }

       return mav;
   }

   @RequestMapping(value = "/loginOut.do", method = RequestMethod.GET)
   public String logOut(SessionStatus sessionStatus) {

       if (!sessionStatus.isComplete())
           sessionStatus.setComplete();

       return "redirect:/loginProcess1.do";
   }
```

실행환경(화면처리) 실습교재

Lab 302 – ajax

eGovFrame

실습 개요

❑ 실습을 통해 AJAX에 대하여 살펴본다.

❑ 실습 순서

- Ajax를 이용한 간단한 구현

1. Ajax controller 작성
2. Ajax view설정

- DB를 연결하여 Ajax의 Autocomplete, AutoSelected기능 구현

1. JSP에 Ajax추가
2. Controller 일부 구현
3. ServiceImpl일부 구현
4. DAO 일부 구현

실행환경(화면처리) 실습교재

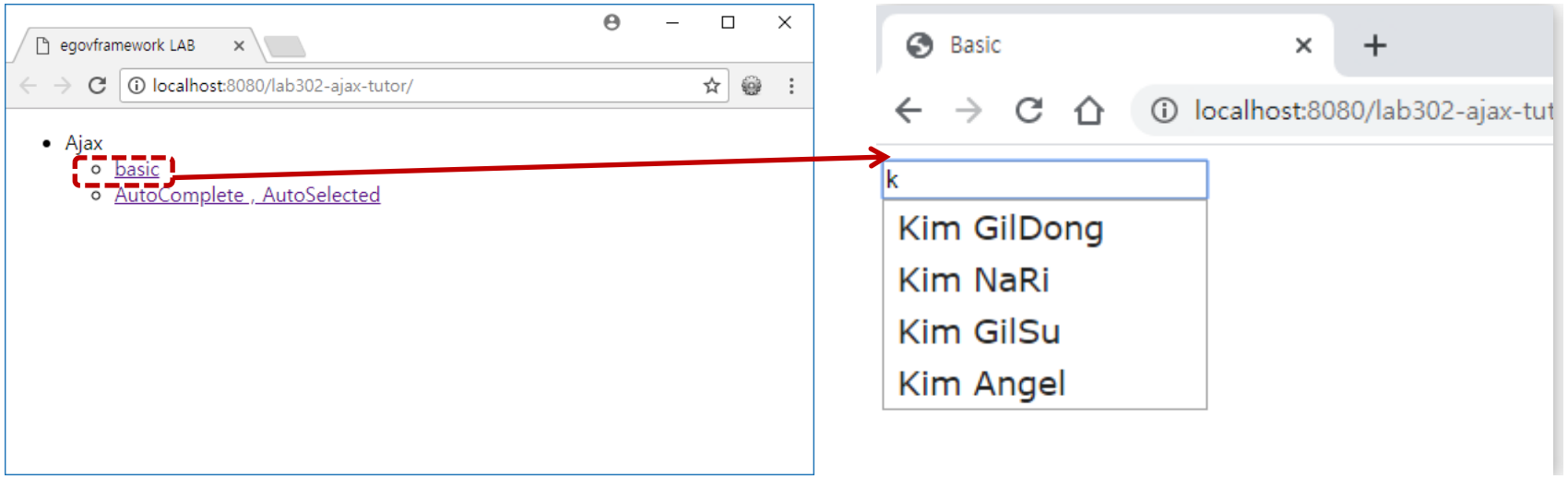
LAB 302-ajax 실습(1)

eGovFrame

WHAT TO DO

'basic' 예제 실행결과 확인

- 프로젝트 선택 마우스 우클릭 > Run As > Run On Server 실행
- 예제 실행 결과 확인 (http://127.0.0.1:8080/lab302-ajax/)



Step 2-1-1. AjaxSimpleController.java 에 simpleAjax 메소드 구현하기

```
@RequestMapping(value = "/autoCompleteSimple.do")
public ModelAndView simpleAjax(@RequestParam("keyword") String keyword) throws Exception{

  ModelAndView modelAndView = new ModelAndView();
  modelAndView.setViewName("jsonView");

  String decode_keyword = URLDecoder.decode(keyword,"utf-8");

  List<?> keywordList = search(decode_keyword);

  LOGGER.debug("result >" + keywordList.toString());

  modelAndView.addObject("resultList", keywordList);

  return modelAndView;
}
```

Step 2-1-2. context-servlet.xml에 MappingJackson2JsonView 빈 등록

```
<bean id="jsonView" class="org.springframework.web.servlet.view.json.MappingJackson2JsonView">
    <property name="contentType" value="text/html;charset=UTF-8"/>
</bean>
```

Step 2-1-3. autocomplete.jsp 에 검색어에 대한 jQuery ajax 자동완성 구현하기

```
$("#keyword").autocomplete({
  source: function(request, response){
    $.ajax({
      url: "<c:url value='/autoCompleteSimple.do'/>",
      contentType: "application/x-www-form-urlencoded; charset=UTF-8",
      data: { keyword: encodeURIComponent(request.term) }, //after the input event
      dataType: "json",
      success: function(returnData, status){
        response(returnData.resultList);
      }
    });
  },
  minLength: 1,
  select: function(event, ui){
    $("#keyword").val(this.value);
  }
});
```

실행환경(화면처리) 실습교재

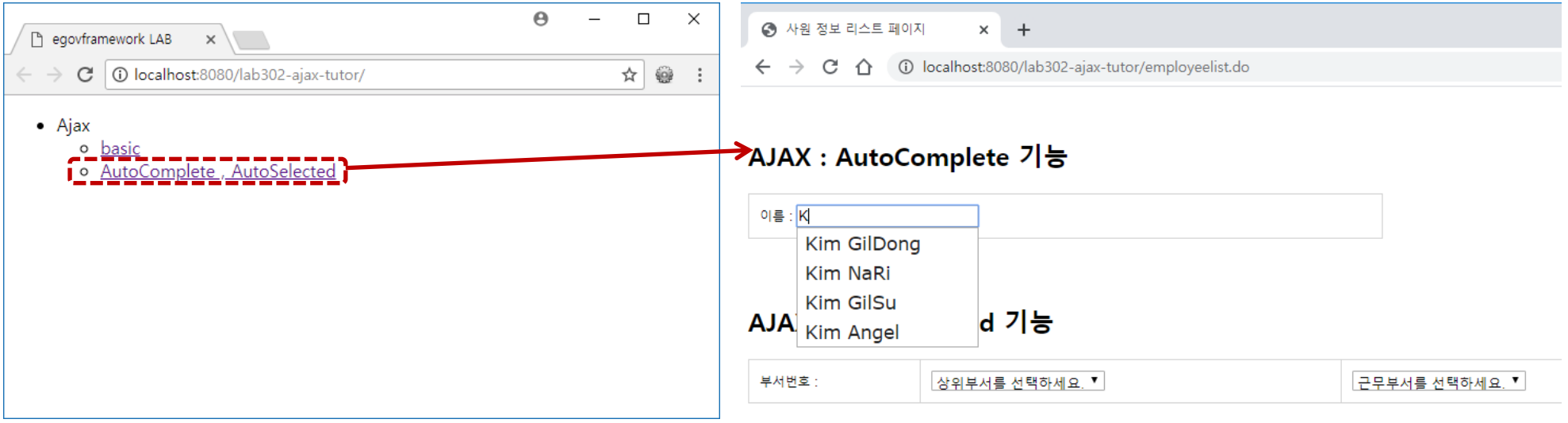
LAB 302-ajax 실습(2)

eGovFrame

WHAT TO DO

‘AutoComplete, AutoSelected’ 예제 실행결과 확인

- 프로젝트 선택 마우스 우클릭 > Run As > Run On Server 실행
- 예제 실행 결과 확인 (http://127.0.0.1:8080/lab302-ajax/)



Step 2-2-1. employeelist.jsp 부서번호에 대한 jQuery ajax call 기능 구현하기

```
$('#superdeptid').change(function(){
  $.ajax({
    url: "<c:url value='/autoSelectDept.do'/>",
    contentType: "application/x-www-form-urlencoded; charset=UTF-8",
    data: {depth:2, superdeptid:encodeURIComponent($('#superdeptid option:selected').val())},
    dataType: "json",
    success: function(returnData, status){
      $('#departmentid').loadSelectDept(returnData,"근무부서를 선택하세요.");
    }
  });
});
```

Step 2-2-2. AjaxController.java에서 employeeService.getNameListForSuggest 메소드를 호출하여 결과를 가져온다.

```
List<String> nameList = employeeService.getNameListForSuggest(searchName);
```

Step 2-2-3. EmployeeService.java의 getNameListForSuggest Interface 메소드 구현하기

```
public List<String> getNameListForSuggest(String namePrefix);
```

Step 2-2-4. EmployeeServiceImpl.java의 getNameListForSuggest 메소드를 구현한다.
(comment 처리를 지운다.)

```
public List<String> getNameListForSuggest(String namePrefix) {

}
```

Step 2-2-5. EmployeeServiceImpl.java의 getNameListForSuggest 메소드에서
employeeDao의 getNameListForSuggest 메소드를 이용하여 검색한 후 결과를 리턴한다.

```
public List<String> getNameListForSuggest(String namePrefix) {
    return employeeDao.getNameListForSuggest(namePrefix);
}
```