**Framework**

|  |
| --- |
| **package**mypack;  **public abstract class** Action {  **public abstract void** execute();    } |

|  |
| --- |
| **package**mypack;  **publicclass**MyAction**extends** Action{  **publicvoid** execute(){  System.*out*.println("my action calisiyor");  }  } |

ApplicationResources.properties

|  |
| --- |
| action= mypack.MyAction |

|  |
| --- |
| import java.util.ResourceBundle;  import mypack.Action;  public class ActionTest {  public static void main(String[] args) {  try {  String actkey = ResourceBundle.getBundle(  "csystem.resource/ApplicationResources").getString("action");  Action act=(Action) Class.forName(actkey).newInstance();  act.execute();  } catch (InstantiationException e) {  // TODO Auto-generated catch block  e.printStackTrace();  } catch (IllegalAccessException e) {  // TODO Auto-generated catch block  e.printStackTrace();  } catch (ClassNotFoundException e) {  // TODO Auto-generated catch block  e.printStackTrace();  }  }  } |

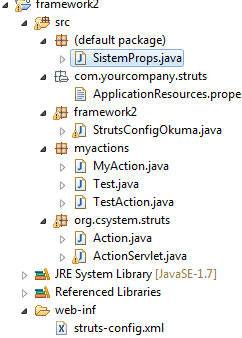
my action calisiyor

**struts-config.xml Okuma**

|  |
| --- |
| <?xmlversion=*"1.0"*encoding=*"UTF-8"*?>  <struts-config>  <data-sources/>  <form-beans/>  <global-exceptions/>  <global-forwards/>  <action-mappings>  <actionpath=*"/my"*type=*"myactions.MyAction"*>  <set-propertyproperty=*"cancellable"*value=*"true"*/>  </action>  </action-mappings>  <message-resourcesparameter=*"com.yourcompany.struts.ApplicationResources"*/>  </struts-config> |

|  |
| --- |
| import java.io.IOException;  import org.jdom.Attribute;  import org.jdom.Document;  import org.jdom.Element;  import org.jdom.JDOMException;  import org.jdom.input.SAXBuilder;  public class StrutsConfigOkuma {  /\*\*  \* @paramargs  \*/  public static void main(String[] args) {  SAXBuildersb = new SAXBuilder();  Document doc;  try {  doc = sb.build("web-inf/struts-config.xml");  Element root = doc.getRootElement();  String actionName = root.getChild("action-mappings")  .getChild("action").getAttribute("type").getValue();    System.out.println(actionName);  } catch (JDOMException e) {  // TODO Auto-generated catch block  e.printStackTrace();  } catch (IOException e) {  // TODO Auto-generated catch block  e.printStackTrace();  }  }  } |

myactions.MyAction

****

|  |
| --- |
| name=ad  age=yas  errormessage=hatali giris |

|  |
| --- |
| **package** org.csystem.struts;  **import** java.io.File;  **import** java.io.FileInputStream;  **import** java.io.IOException;  **import** java.util.Properties;  **import** java.util.Set;  **import** org.jdom.Document;  **import** org.jdom.Element;  **import** org.jdom.JDOMException;  **import** org.jdom.input.SAXBuilder;  **import** myactions.MyAction;  **public** **class** ActionServlet {  **public** ActionServlet() {  // **TODO** Auto-generated constructor stub  **try** {  SAXBuilder sb = **new** SAXBuilder();  Document doc;  doc = sb.build("web-inf/struts-config.xml");  Element root = doc.getRootElement();  String actionName = root.getChild("action-mappings")  .getChild("action").getAttribute("type").getValue();  Class cls = Class.*forName*(actionName);  Action my = (Action) cls.newInstance();  my.execute();  String messageResources = root.getChild("message-resources")  .getAttribute("parameter").getValue();  System.*out*.println(messageResources);  String path = messageResources.replace(".", "\\");    FileInputStream fi=**new** FileInputStream(System.*getProperty*("user.dir")+File.*separator*+"src"+File.*separator*+path+".properties");  Properties props=**new** Properties();  props.load(fi);  Set<Object> keys = props.keySet();  **for** (Object key : keys) {  System.*out*.println(key+"\t"+props.getProperty((String)key));  }  } **catch** (ClassNotFoundException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  } **catch** (InstantiationException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  } **catch** (IllegalAccessException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  } **catch** (JDOMException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  } **catch** (IOException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  }  }  } |

**BirSınıftakiMetodlarınBulunması (reflection)**

|  |
| --- |
| **publicclass**MyAction {  **private** String ad;  **private** String soyad;  **public** String getSoyad() {  **return**soyad;  }  **publicvoid**setSoyad(String soyad) {  **this**.soyad = soyad;  }  **publicvoid**setAd(String ad) {  System.*out*.println(ad);  **this**.ad=ad;  }  **public** String getAd() {  **return**ad;  }  } |

|  |
| --- |
| **import**java.lang.reflect.Method;  **publicclass**MetodOkuma {  /\*\*  \* **@param**args  \*/  **publicstaticvoid** main(String[] args) {  **try** {  Classcls = Class.*forName*("MyAction");  Method[] methods = cls.getMethods();  **for** (**int**i = 0; i<methods.length; i++) {  System.*out*.println(methods[i].getName());  }  } **catch** (ClassNotFoundException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  }  }  } |

getSoyad

setSoyad

setAd

getAd

getClass

hashCode

equals

toString

notify

notifyAll

wait

wait

wait

**BirSınıftakiMetodlarınFarklıŞekildeÇalıştırılması**

Özelliklebelirlibirşartabağlımetodlarınçalıştırılmasındakullanılabilir.Örneğin “set” ilebaşlayanmetodlarınçalıştırılmasıistenebilir.

|  |
| --- |
| **import**java.lang.reflect.InvocationTargetException;  **import**java.lang.reflect.Method;  **publicclass**MetodCalistirma {  /\*\*  \* **@param**args  \*/  **publicstaticvoid** main(String[] args) {  Classcls;  **try** {  cls = Class.*forName*("MyAction");  MyAction act = (MyAction) (cls.newInstance());  Method[] methods = cls.getMethods();  **for** (**int**i = 0; i<methods.length; i++) {  Method meth = methods[i];  **if** (meth.getName().startsWith("set"))  meth.invoke(act, "deneme");  }  } **catch** (ClassNotFoundException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  } **catch** (InstantiationException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  } **catch** (IllegalAccessException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  } **catch** (IllegalArgumentException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  } **catch** (InvocationTargetException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  }  }  } |

|  |
| --- |
| **package**org.csystem.struts.action;  **publicabstractclass** Action {  **publicabstractvoid** execute();  } |

|  |
| --- |
| package org.csystem.struts.action;  import java.io.IOException;  import java.util.ResourceBundle;  import org.jdom.Document;  import org.jdom.Element;  import org.jdom.JDOMException;  import org.jdom.input.SAXBuilder;  public class ActionServlet {  public ActionServlet() {  }  public void execute() {  SAXBuildersb = new SAXBuilder();  Document doc;      try {  doc = sb.build("web-inf/struts-config.xml");  Element root = doc.getRootElement();  String actionName = root.getChild("action-mappings")  .getChild("action").getAttribute("type").getValue();  Action action = (Action) Class.forName(actionName).newInstance();  action.execute();  } catch (InstantiationException e) {  // TODO Auto-generated catch block  e.printStackTrace();  } catch (IllegalAccessException e) {  // TODO Auto-generated catch block  e.printStackTrace();  } catch (ClassNotFoundException e) {  // TODO Auto-generated catch block  e.printStackTrace();  } catch (JDOMException e) {  // TODO Auto-generated catch block  e.printStackTrace();  } catch (IOException e) {  // TODO Auto-generated catch block  e.printStackTrace();  }  }  } |

|  |
| --- |
| **package**myactions;  **import**org.csystem.struts.action.Action;  **publicclass**MyAction**extends** Action {    **publicvoid** execute() {  System.*out*.println("my action execute");  }  } |

|  |
| --- |
| import org.csystem.struts.action.ActionServlet;  import org.csystem.struts.action.ActionServletProps;  public class Test {  /\*\*  \* @paramargs  \*/  public static void main(String[] args) {  new ActionServlet().execute();  }  } |

**Servlet’li Framework Örneği (Struts’abenzer)**



|  |
| --- |
| package csystem.action;  import javax.servlet.http.HttpServletRequest;  import javax.servlet.http.HttpServletResponse;  import csystem.form.PersonelForm;  import framework.Action;  import framework.ActionForm;  public class PersonelAction extends Action {    public void execute(ActionForm form, HttpServletRequest request,  HttpServletResponse response) {  PersonelForm per=(PersonelForm)form;  System.out.println(per.getAd());  System.out.println(per.getSoyad());  System.out.println(per.getEmail());    }  } |

|  |
| --- |
| **package**csystem.form;  **import**framework.ActionForm;  **publicclass**PersonelForm**extends**ActionForm{  **private** String ad;  **private** String soyad;  **private** String email;  **public** String getAd() {  **return**ad;  }  **publicvoid**setAd(String ad) {  **this**.ad = ad;  }  **public** String getSoyad() {  **return**soyad;  }  **publicvoid**setSoyad(String soyad) {  **this**.soyad = soyad;  }  **public** String getEmail() {  **return**email;  }  **publicvoid**setEmail(String email) {  **this**.email = email;  }  } |

|  |
| --- |
| action1=csystem.action.PersonelAction  form1=csystem.form.PersonelForm |

|  |
| --- |
| package framework;  import javax.servlet.http.HttpServletRequest;  import javax.servlet.http.HttpServletResponse;  public abstract class Action {  public abstract void execute(ActionForm form,  HttpServletRequest request, HttpServletResponse response);  } |

|  |
| --- |
| **package** framework;  **publicclass**ActionForm {    } |

|  |
| --- |
| package framework;  import java.io.IOException;  import java.lang.reflect.InvocationTargetException;  import java.lang.reflect.Method;  import java.util.Enumeration;  import java.util.HashMap;  import java.util.ResourceBundle;  import java.util.Set;  import javax.servlet.ServletException;  import javax.servlet.http.HttpServlet;  import javax.servlet.http.HttpServletRequest;  import javax.servlet.http.HttpServletResponse;  public class ActionServlet extends HttpServlet {  public ActionServlet() {  }// constr  protected void doGet(HttpServletRequestreq, HttpServletResponseresp)  throws ServletException, IOException {  fonk(req, resp);  }  protected void doPost(HttpServletRequestreq, HttpServletResponseresp)  throws ServletException, IOException {  // TODO Auto-generated method stub  fonk(req, resp);  }  private void fonk(HttpServletRequestreq, HttpServletResponseresp) {  Enumeration params = req.getParameterNames();  String actkey = ResourceBundle.getBundle(  "csystem.resource/ApplicationResources").getString("action1");  String formkey = ResourceBundle.getBundle(  "csystem.resource/ApplicationResources").getString("form1");  try {  Action action = (Action) Class.forName(actkey).newInstance();  ActionForm form = (ActionForm) Class.forName(formkey).newInstance();  Class cls = form.getClass();  Method[] methods = cls.getMethods();  int index = 0;  HashMap<String, Method>setMeths = new HashMap<String, Method>();  for (inti = 0; i<methods.length; i++) {  Method method = methods[i];  if (method.getName().startsWith("set")) {  String str = method.getName().substring(3, 4).toLowerCase()  .concat(method.getName().substring(4));  setMeths.put(str, method);  }// if  }// for  while (params.hasMoreElements()) {  String temp = (String) params.nextElement();  Set<String>methKeys = setMeths.keySet();  for (String metName : methKeys) {  if (metName.equals(temp)) {  setMeths.get(metName).invoke(form,  req.getParameter(temp));  System.out.println(temp);  }  }  }  action.execute(form, req, resp);  } catch (InstantiationException e) {  // TODO Auto-generated catch block  e.printStackTrace();  } catch (IllegalAccessException e) {  // TODO Auto-generated catch block  e.printStackTrace();  } catch (ClassNotFoundException e) {  // TODO Auto-generated catch block  e.printStackTrace();  } catch (IllegalArgumentException e) {  // TODO Auto-generated catch block  e.printStackTrace();  } catch (InvocationTargetException e) {  // TODO Auto-generated catch block  // e.printStackTrace();  } catch (SecurityException e) {  // TODO Auto-generated catch block  e.printStackTrace();  }  }  } |

|  |
| --- |
| <html>  <body>  <formaction=*"servlet/ActionServlet"*>  Ad : <inputtype=*"text"*name=*"ad"*><br>  Soyad : <inputtype=*"text"*name=*"soyad"*><br>  Email : <input type=*"text"*name=*"email"*><br>  <inputtype=*"submit"*value=*"gonder"*/>  </form>  </body>  </html> |