* 学习目标

 * 能够掌握登录的案例

    * Lombok,MVC,Web的三层架构

    * bean,dao(impl),service(impl),controller

 * 能够掌握Servlet3.0的使用

    * @WebServlet,@WebInitParams

 * 能够理解请求和响应的概述

    * Request --- Response

 * 能够掌握Request

    * ServletRequest,HttpServletReuest

 * 能够编写注册案例

 * 能够掌握Repsonse

---

* 能够掌握登录的案例

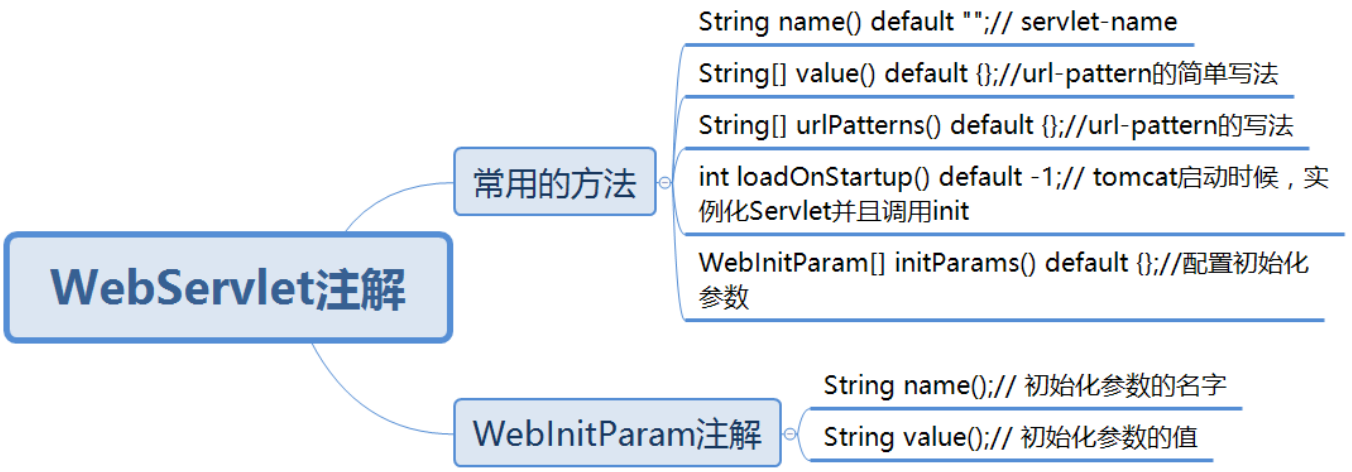
```
* 数据库：mysql-lg01-t_user(id,username,password)
* 代码层次
  * com.lg.bean
    * User
  * com.lg.dao
    * UserDao
     * UserDaoImpl
  * com.lg.service
    * UserService
     * UserServiceImpl
  * com.lg.servlet(controller)
    * LoginServlet
  * 页面：login.html,success.html,fail.html
 * MVC 模式：Model-View-Controller
    * 用户发送请求-到控制层（接受用户的数据，调用业务逻辑方法，根据放回结果跳转到不同页
 * Web三层架构
```

```
    * 数据持久层，Web层次（业务逻辑层），页面层

* 代码
@Data
@AllArgsConstructor
@NoArgsConstructor
public class User {
    private int id;
    private String username;
    private String password;

}
public interface UserDao {
     User getUser(String username, String password);
}
public class UserDaoImpl implements UserDao {
    @Override
    public User getUser(String username, String password)  {
        String sql="SELECT * FROM t_user WHERE username=? AND password=? ";
        QueryRunner runner=new QueryRunner(DataSourceUtils.getDataSource());
        try {
            User user = runner.query(sql, new BeanHandler<User>(User.class), us
            return user;
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return null;
    }
}

public interface LoginService {
    boolean login(String username,String password);

    User loginUser(String username, String password);
}

public class LoginServiceImpl implements LoginService {
    private UserDao mUserDao;
    public LoginServiceImpl(){
        mUserDao=new UserDaoImpl();
```

```
57      }
58      @Override
59      public boolean login(String username, String password) {
60          User user = loginUser(username,password);
61          return user!=null;
62      }
63
64      @Override
65      public User loginUser(@NonNull String username, @NonNull String password) {
66          User user = mUserDao.getUser(username, password);
67          return user;
68      }
69  }
70
71  @WebServlet(value = "/login")
72  public class LoginServlet extends HttpServlet {
73      @Override
74      protected void doGet(HttpServletRequest req, HttpServletResponse resp) thro
75          // 1 获取用户从页面输入的数据：例如表单提交的数据
76          String username = req.getParameter("username");
77          String password = req.getParameter("password");
78          System.out.println("username = " + username);
79          System.out.println("password = " + password);
80
81          // 2 调用登录业务--调用dao--查询数据库
82          LoginService loginService=new LoginServiceImpl();
83          boolean isOk = loginService.login(username, password);
84          if(isOk){
85              getServletContext().setAttribute("msg","login success");
86              getServletContext().getRequestDispatcher("/success").forward(req,re
87          }else{
88              getServletContext().setAttribute("msg","login fail");
89              getServletContext().getRequestDispatcher("/fail").forward(req,resp)
90          }
91      }
92
93      @Override
94      protected void doPost(HttpServletRequest req, HttpServletResponse resp) thr
95          doGet(req,resp);
96      }
```

```java
}
@WebServlet(value = "/success")
public class SuccessServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) thro
        // 获得信息，响应给页面
        String msg = (String) getServletContext().getAttribute("msg");
        resp.getWriter().write(msg);
    }

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) thr
       doGet(req, resp);
    }
}

@WebServlet(value = "/fail")
public class FailServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) thro
        // 获得信息，响应给页面
        String msg = (String) getServletContext().getAttribute("msg");
        resp.getWriter().write(msg);
    }

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) thr
       doGet(req, resp);
    }
}


```

* 能够掌握Servlet3.0的使用

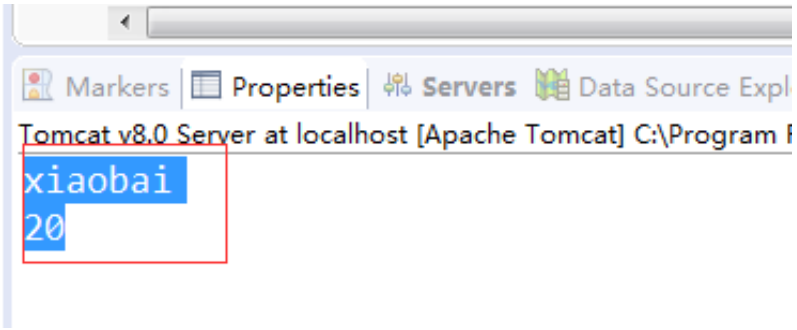



```
案例一：
@WebServlet(name="hello",value= {"/hello"})
public class HelloServlet extends HttpServlet{

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) thro
        resp.getWriter().append("Hello Servlet3.0");
    }

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) thr
    }
}

案例二：
@WebServlet(name="hello",urlPatterns= {"/hello"})
public class HelloServlet extends HttpServlet

案例三：
@WebServlet(name="hello",value= {"/hello"},loadOnStartup=1)
public class HelloServlet extends HttpServlet{

    @Override
    public void init(ServletConfig config) throws ServletException {
        super.init(config);
        System.out.println("HelloServlet init 3.00000000");
    }
```

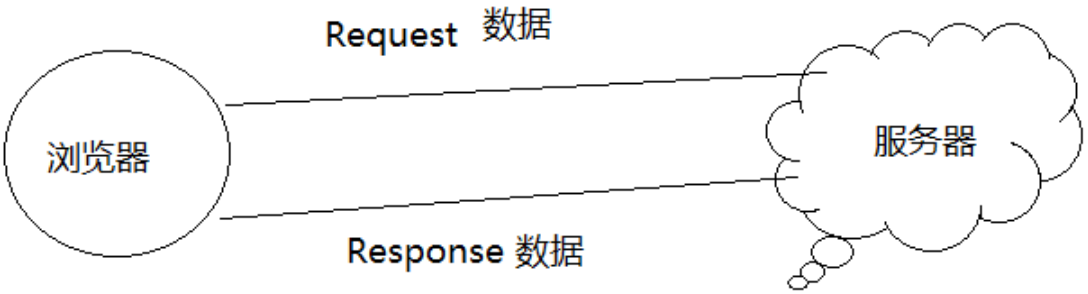
```
28      @Override
29      protected void doGet(HttpServletRequest req, HttpServletResponse resp) thro
30          resp.getWriter().append("Hello Servlet3.0");
31      }
32
33      @Override
34      protected void doPost(HttpServletRequest req, HttpServletResponse resp) thr
35      }
36  }
37
```

* 案例四

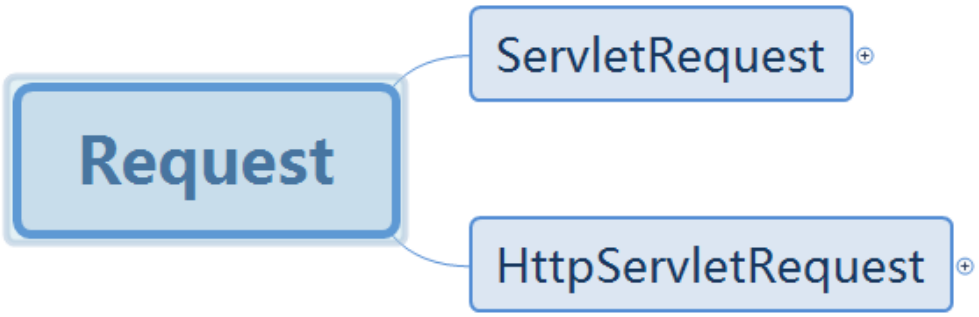
```
@WebServlet(name = "hello", value = { "/hello" }, loadOnStartup = 1, initParams = {
        @WebInitParam(name = "name", value = "xiaobai"),@WebInitParam(name="age",value="20") })
public class HelloServlet extends HttpServlet {

    @Override
    public void init(ServletConfig config) throws ServletException {
        super.init(config);
        System.out.println("HelloServlet init 3.00000000");
    }

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOExce
        String name=getServletConfig().getInitParameter("name");
        String age=getInitParameter("age");
        System.out.println(name);
        System.out.println(age);
```



* 能够理解请求和响应的概述



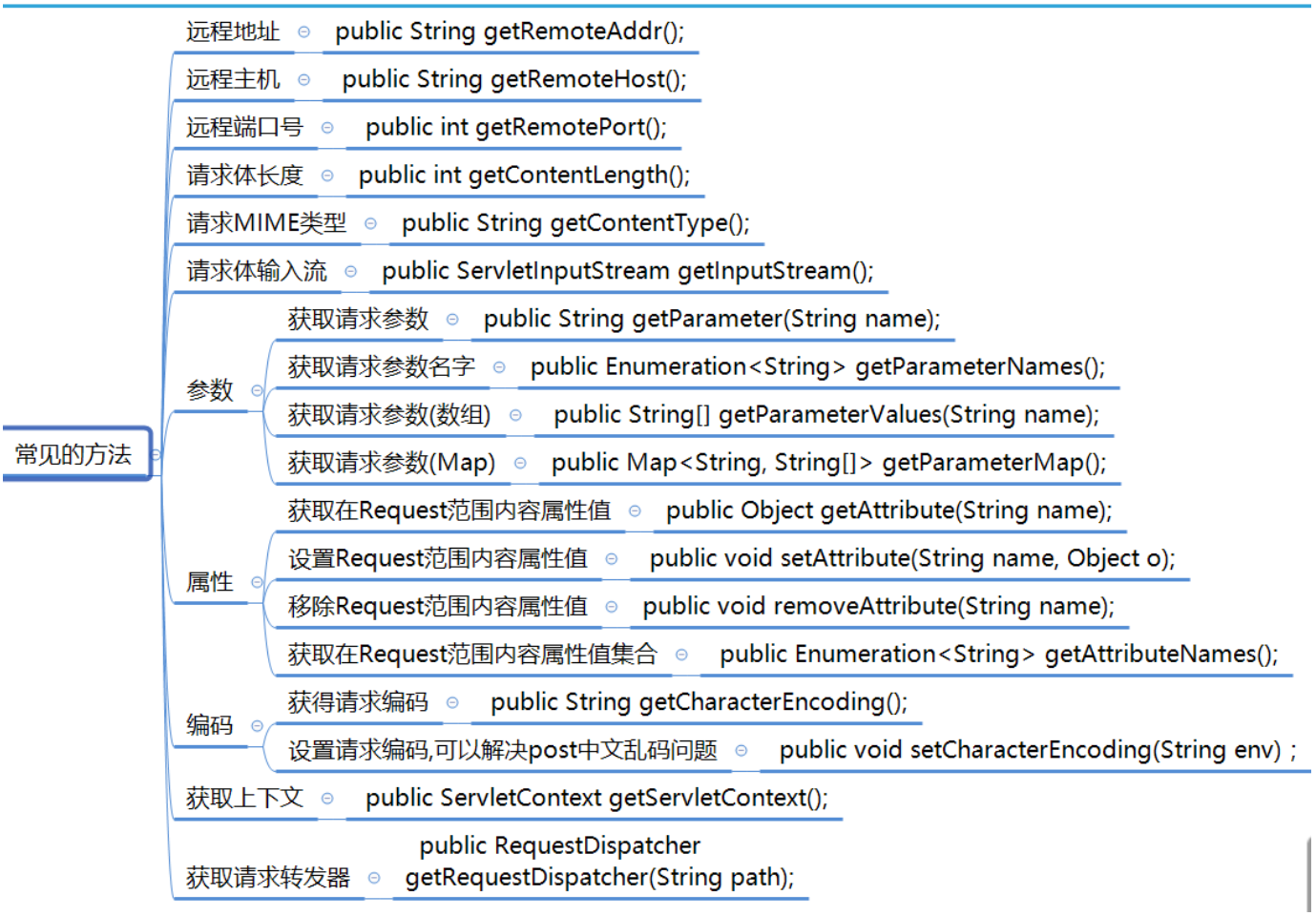
* 能够掌握Request



* ServletRequest

  * 概述

  * 常见方法



* HttpServletRequest


```
* 案例一
  protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws
        String remoteAddr = req.getRemoteAddr();
        String remoteHost = req.getRemoteHost();
        int remotePort = req.getRemotePort();
```

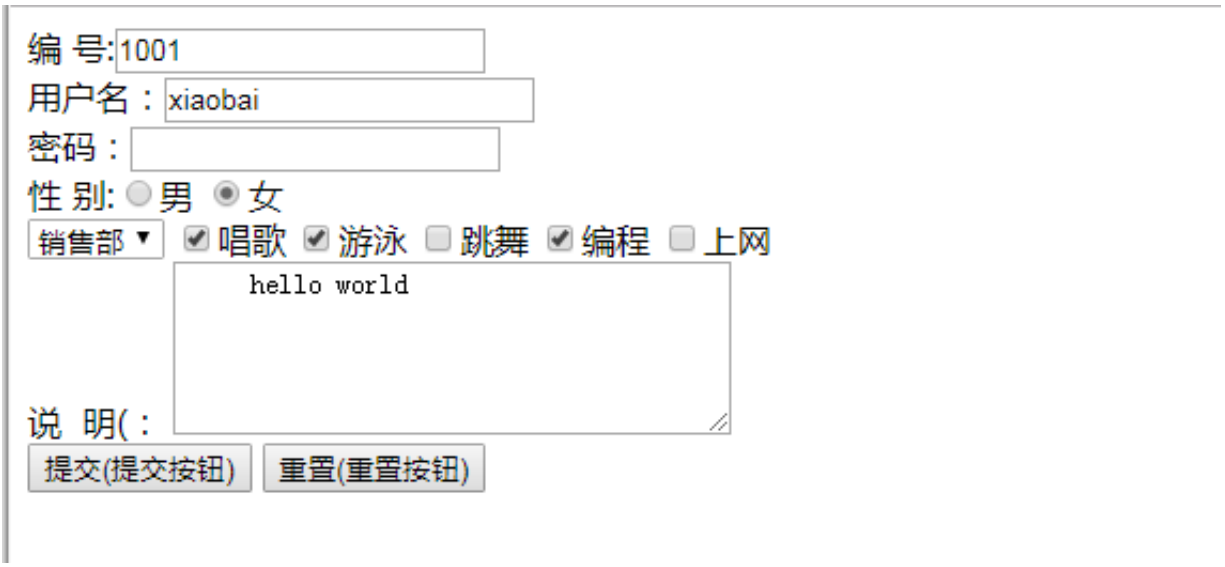
```
        System.out.println("remoteAddr = " + remoteAddr);
        System.out.println("remoteHost = " + remoteHost);
        System.out.println("remotePort = " + remotePort);

        int contentLength = req.getContentLength();
        System.out.println("contentLength = " + contentLength);
        String contentType = req.getContentType();
        System.out.println("contentType = " + contentType);
//        ServletInputStream is = req.getInputStream();
//        byte[] b=new byte[1024];
//        int len=is.read(b);
//        String msg=new String(b,0,len);
//        System.out.println("msg = " + msg);

        String username = req.getParameter("username");
        String password = req.getParameter("password");
        System.out.println("username = " + username);
        System.out.println("password = " + password);
//        String[] hobbies = req.getParameterValues("hobby");
//        for (String hobby : hobbies) {
//            System.out.println("hobby = " + hobby);
//        }
        Enumeration<String> names = req.getParameterNames();
        while (names.hasMoreElements()){
            System.out.println(names.nextElement());
        }
        Map<String, String[]> map = req.getParameterMap();
        Set<Map.Entry<String, String[]>> entries = map.entrySet();
        for(Map.Entry<String, String[]> entry:entries){
            System.out.println(entry.getKey()+":"+entry.getValue()[0]);
        }
    }
* 案例二:
public class MBeanUtils {
    public static void populate(HttpServletRequest req,Object obj){
        Map<String,String> map=new HashMap<String,String>();
        Enumeration<String> names = req.getParameterNames();
        while(names.hasMoreElements()){
            String name=names.nextElement();
            String value=req.getParameter(name);
```

```
            map.put(name,value);
        }
        try {
            BeanUtils.populate(obj,map);
        } catch (IllegalAccessException e) {
            e.printStackTrace();
        } catch (InvocationTargetException e) {
            e.printStackTrace();
        }
    }
}

 @Override
protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws S
        User user=new User();
        MBeanUtils.populate(req,user);
        System.out.println("user = " + user);
 }


* 案例三：
  @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) thro
        User user=new User();
        MBeanUtils.populate(req,user);
        System.out.println("user = " + user);
        boolean isRegOk=true;
        if(isRegOk){
            req.setAttribute("msg","reg success");
            req.getRequestDispatcher("/success").forward(req,resp);
        }else{
            req.setAttribute("msg","reg fail");
            req.getRequestDispatcher("/fail").forward(req,resp);
        }
    }
* 案例四：
  @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) thro
//        String header = req.getHeader("Accept-Encoding");
        Enumeration<String> headerNames = req.getHeaderNames();
```

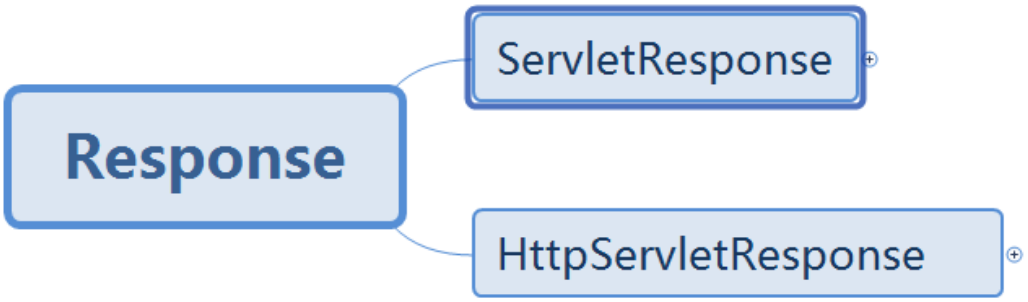
```
86          while(headerNames.hasMoreElements()){
87              String name=headerNames.nextElement();
88              String value=req.getHeader(name);
89              System.out.println(name+":"+value);
90          }
91  //          System.out.println("header = " + header);
92          System.out.println(req.getMethod());
93          System.out.println(req.getRequestURL());
94          System.out.println(req.getRequestURI());
95          System.out.println(req.getContextPath());
96          System.out.println(req.getQueryString());
97      }
```

* 能够编写注册案例

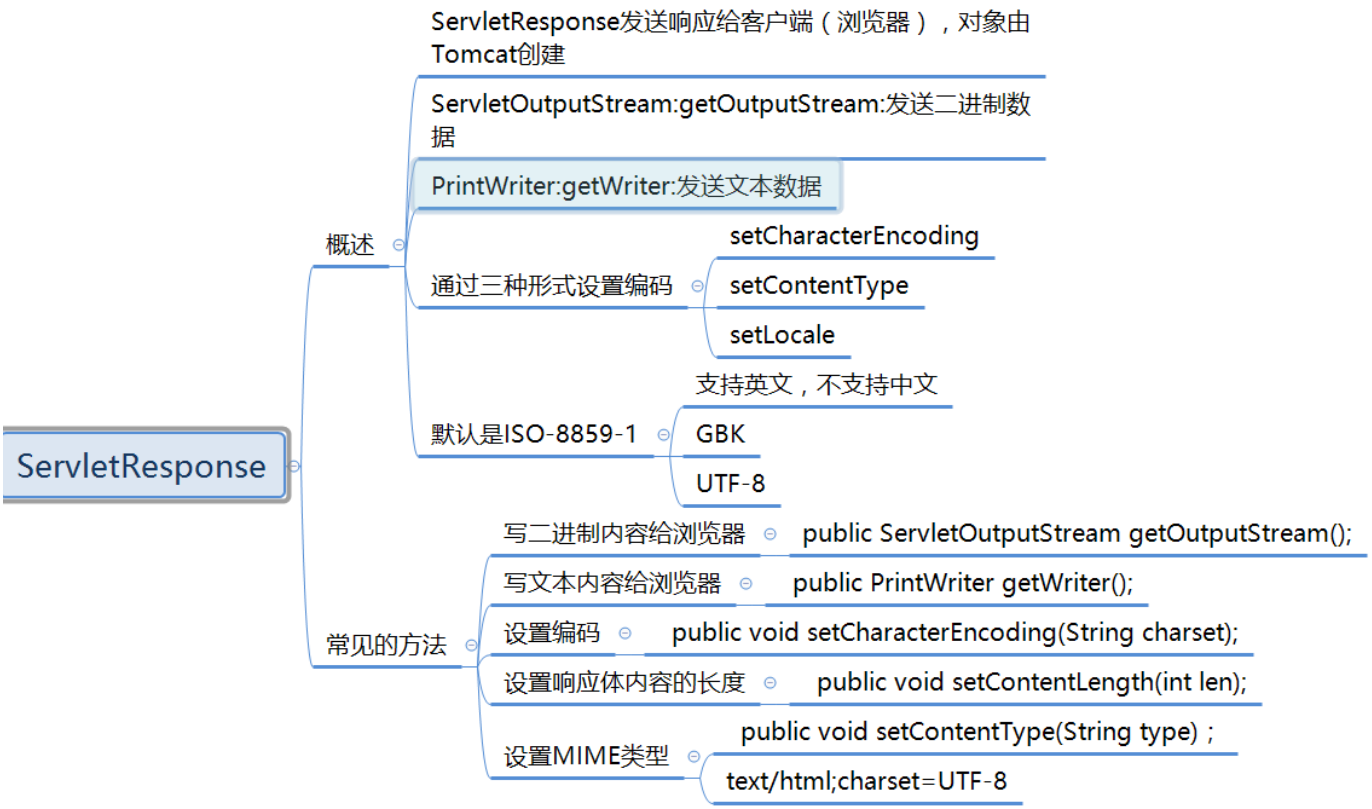
  * 作业



* 能够掌握Repsonse



* ServletResponse



* HttpServletResponse


```
案例一：
  @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) thro
            // ISO-8859-1
//        resp.setCharacterEncoding("GBK");
        resp.setContentType("text/html;charset=UTF-8");
        resp.getWriter().write("亮哥教育，做教育我们是认真的");
    }
```