
DataArt

DataArt

Java-интенсив

Chapter 6.1

DataArt

MVC – Model View Controller

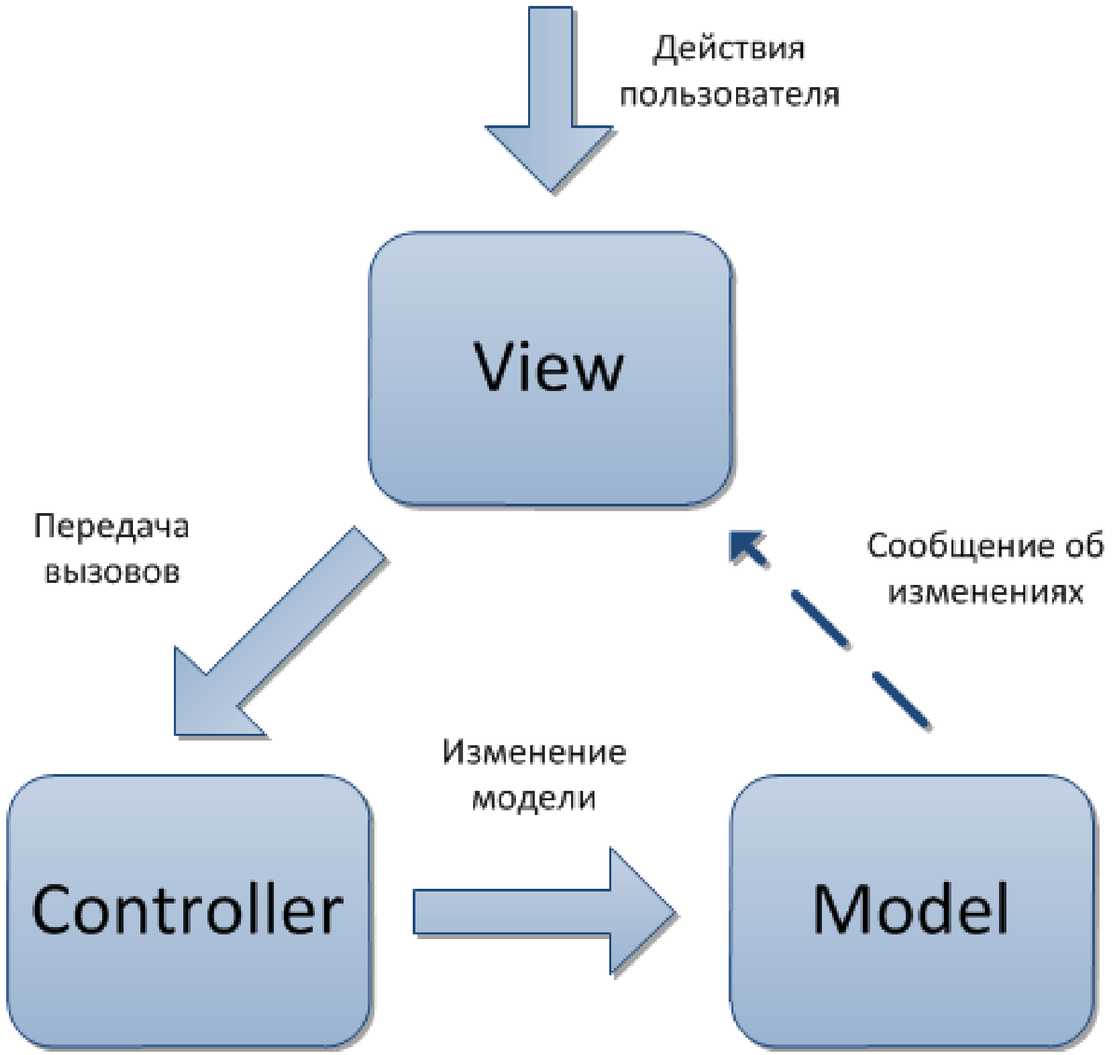
MVC

DataArt

- В больших приложениях часто становится много классов
- Если у этих классов нет четкой структуры, в них очень сложно разбираться
- MVC = Model-View-Controller, паттерн проектирования
- Фактически – это способ разделять классы приложений на слои
- Каждый слой должен общаться только с предыдущим

MVC

DataArt


Действия
пользователя
View
Передача
вызовов
Сообщение об
изменениях
Изменение
модели
Controller
Model

Основные слои приложения - View.

DataArt

- View – слой того, что видит пользователь. Тут находится все, что отвечает за внешний вид приложения.
- View – layer должен практически ничего не знать о какой либо логике приложения.
- Например, есть у вас список пользователей – друзей в соц.сети. View должно их красиво нарисовать, и все.
- View layer не должен знать, каким именно образом был сформирован этот список

Основные слои приложения - Controller.

DataArt

- Controller получает информацию о том, что пользователь хочет увидеть какую-либо view или выполнить какое-либо действие.
- В концепции MVC, контроллер прячет за собой основную логику приложения
- Фактически, controller чаще всего получает данные со view
- Например, view это страница логина (с полями логина и пароля)
- View отправит данные в контроллер, который решит, что показать дальше пользователю (например, ошибку, или домашнюю страницу)

Основные слои приложения - Model.

DataArt

- Слой Model в MVC, это бизнес-обьекты (например User – class)
- Приложение манипулирует этими обьектами, это основной элемент бизнес-логики

DataArt

Chapter 6.2

Слои в Java-приложениях

Основные слои приложения – бизнес-логика (service layer).

DataArt

- Это место, где сосредоточена основная логика приложения.
- Например, контроллер прислал нам логин и пароль, и спрашивает, корректные ли они.
- Уровень бизнес-логики должен как раз проверить, корректны ли логин и пароль (например, просто методом equals, это будет основная задача класса LoginService)
- Вот только Service – layer должен откуда-то получить данные (по логину пользователя, например)

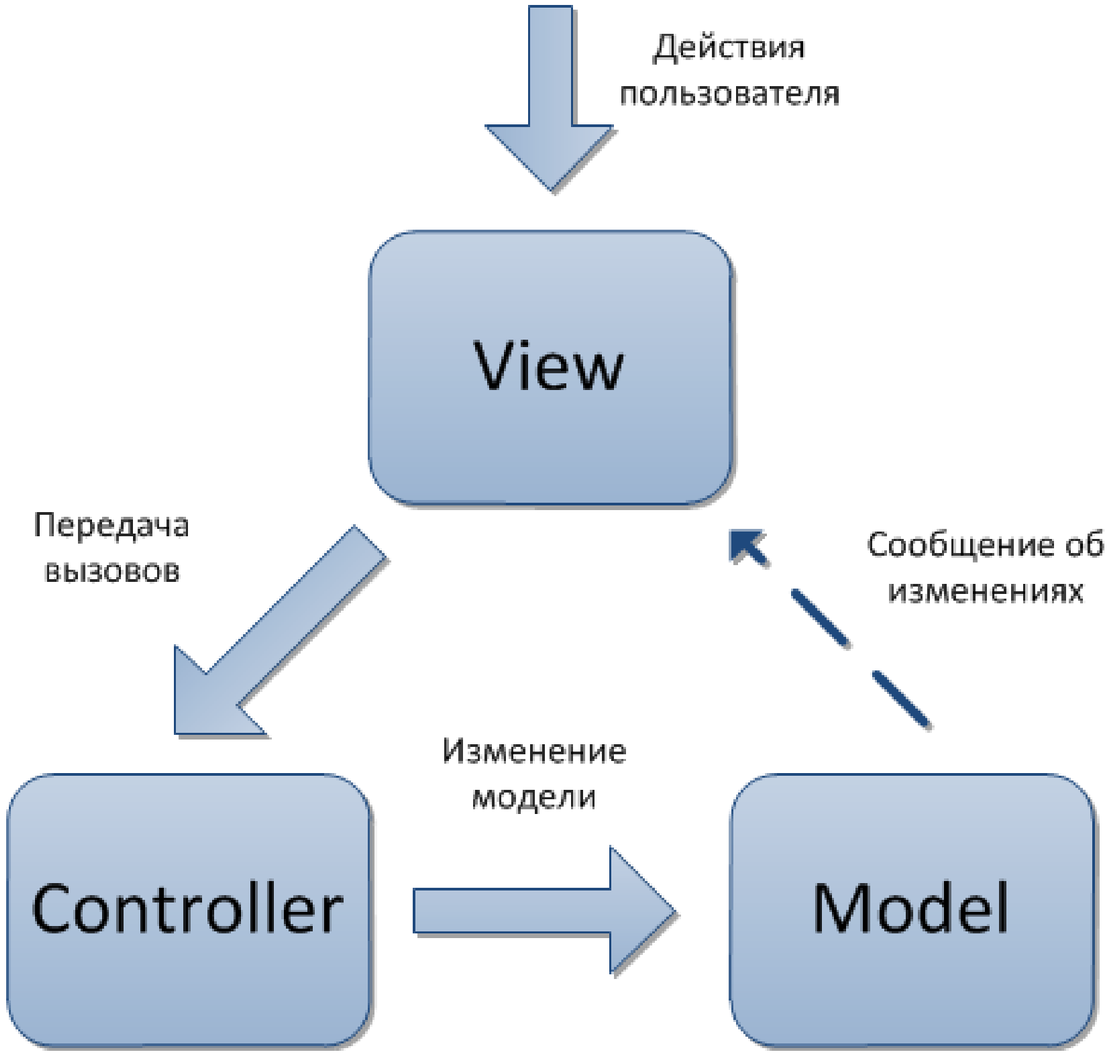
Основные слои приложения – DataAccess

DataArt

- Это набор классов, которые предоставляют данные (обычно бизнес-логика данных не проверяется)
- Например, на вход метода приходит логин, а DAO/Repository class должен вернуть данные об этом пользователе
- DataAccess может читать из большого количества разных источников (самый простой – файлы, самый распространенный – SQL базы данных)

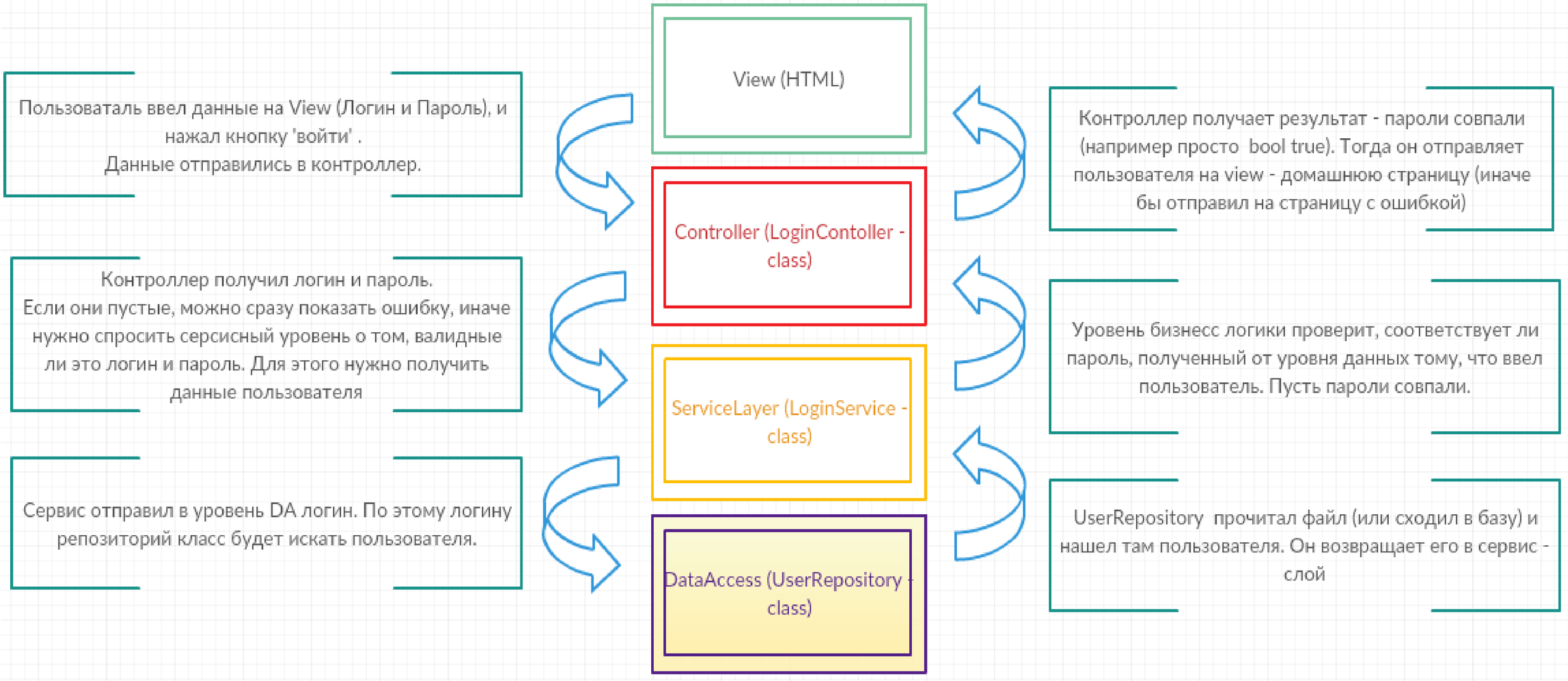
Еще раз посмотрим на схему MVC

DataArt



Теперь посмотрим, как слои выглядят чаще всего в приложении

DataArt



DataArt

Вопросы и ответы

DataArt

Chapter 6.3

Tomcat и первое web-приложение

Tomcat

- Мы будем создавать специальные классы – сервлеты.
- Servlet мы будем в основном использовать как Controller.
- Для View – мы будем использовать JSP – (java server pages). Фактически позволяет совмещать html и java-code.
- Данные будем получать просто из текстовых файлов (но для начала просто из заранее подготовленных java-коллекций)
- Все это будем помещать внутрь сервлет – контейнера Tomcat

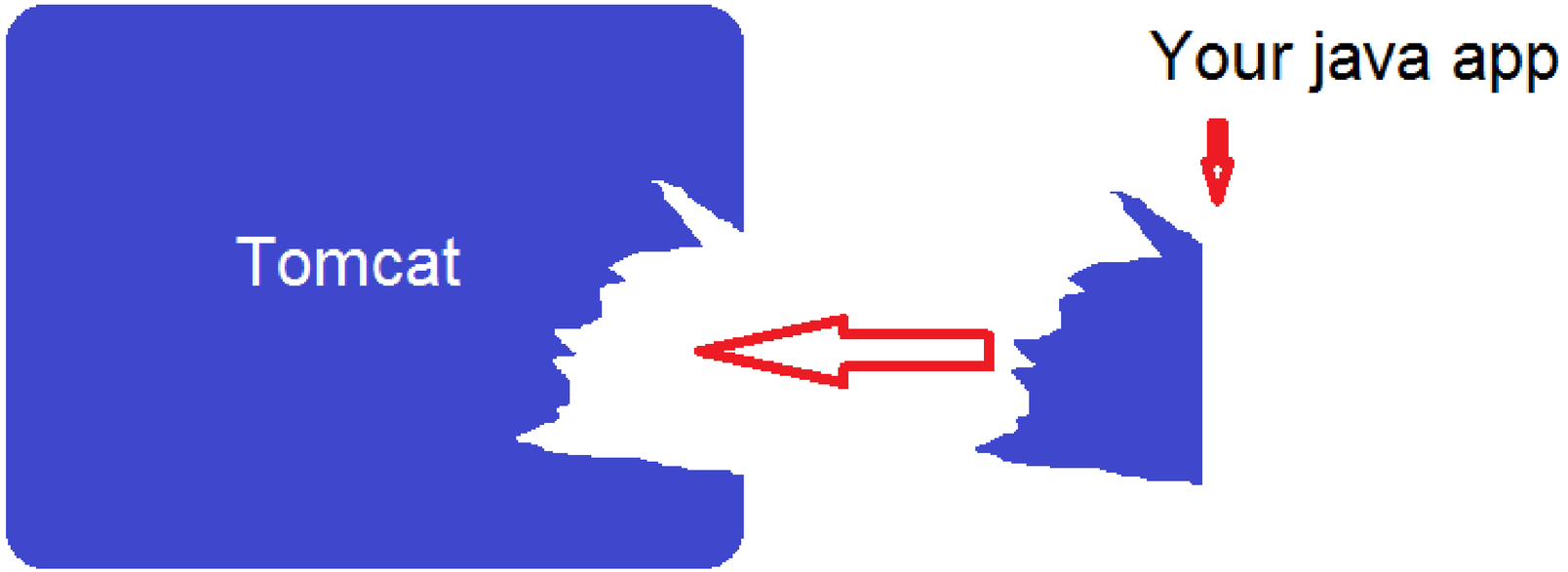
Что такое Tomcat?

DataArt

- Tomcat – сервлет-контейнер
- То есть он понимает специальный формат классов – сервлеты (и еще несколько, в том числе jsp)
- В целом, томкат - это приложение на java. Чтобы интегрировать в него свои классы, надо положить скомпилированные классы в определенную папку (можно настроить idea делать это автоматически)

Что такое Tomcat?

DataArt



Пример сервлета

DataArt

- Специальный класс, унаследованный от HTTP сервлет
- Может обрабатывать HTTP запросы (GET, POST и др)
- Можно конфигурировать его аннотациями или через специальный xml файл

Что такое сервлет

DataArt

```
2
3
4   import javax.servlet.ServletException;
5   import javax.servlet.http.HttpServlet;
6   import javax.servlet.http.HttpServletRequest;
7   import javax.servlet.http.HttpServletResponse;
8   import java.io.IOException;
9
10  public class HelloWorldController extends HttpServlet {
11
12      @Override
13      protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
14          String username = req.getParameter("userName");
15          resp.getWriter().write("hello, " + username);
16      }
17
18  }
19
```

Pom.xml example

DataArt

servletexample

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>com.kondusov</groupId>
    <artifactId>servletexample</artifactId>
    <version>1.0-SNAPSHOT</version>

    <dependencies>
        <dependency>
            <groupId>javax.servlet</groupId>
            <artifactId>javax.servlet-api</artifactId>
            <version>3.0.1</version>
            <scope>provided</scope>
        </dependency>
        <dependency>
            <groupId>javax.servlet.jsp</groupId>
            <artifactId>jsp-api</artifactId>
            <version>2.2</version>
            <scope>provided</scope>
        </dependency>
    </dependencies>

</project>
```

Pom.xml example

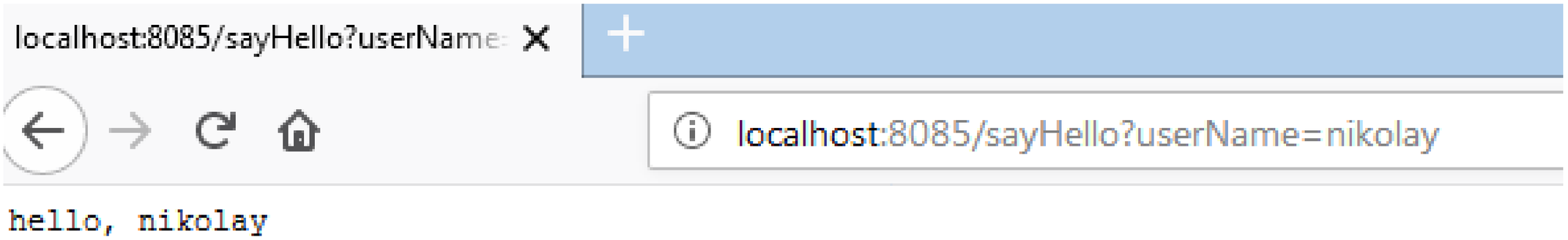
DataArt

```xml
<build>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-compiler-plugin</artifactId>
            <version>3.6.1</version>
            <configuration>
                <source>1.8</source>
                <target>1.8</target>
            </configuration>
        </plugin>
    </plugins>
</build>
```

Пример сервлета

DataArt

- Скомпилируем проект
- Задеплоим его в томкат
- Проверим результат



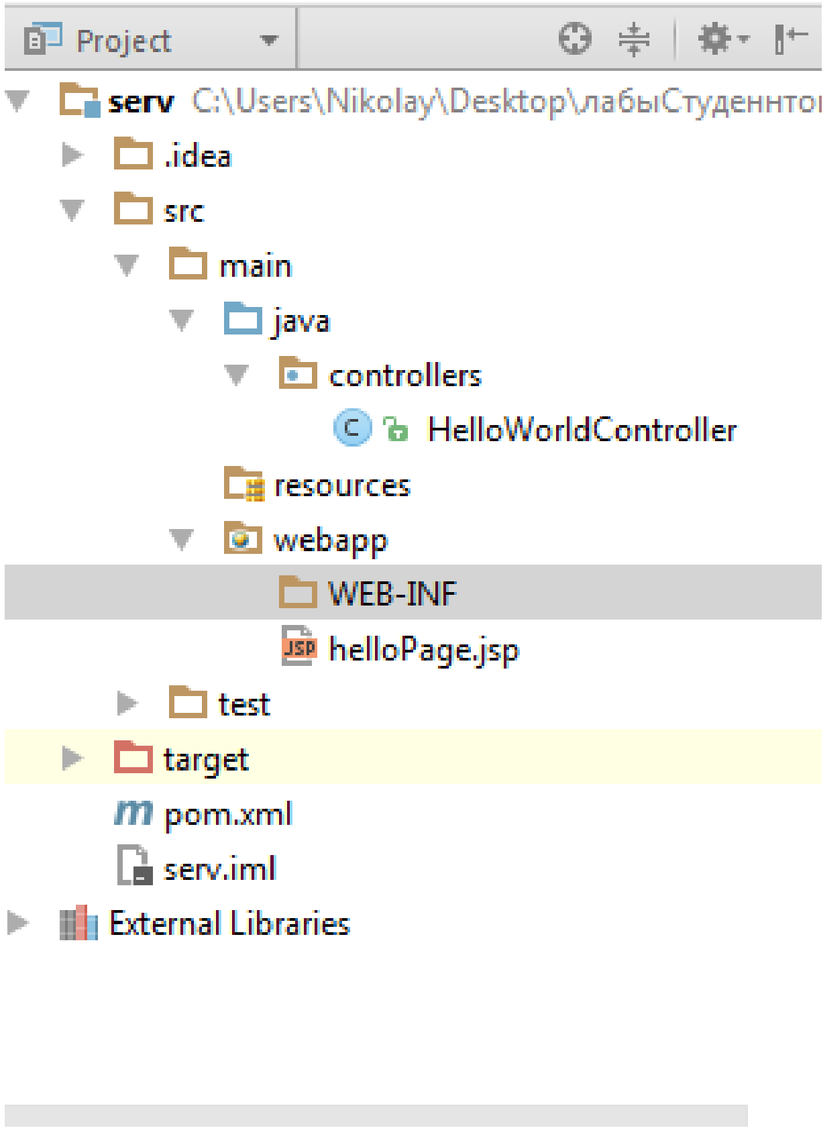
Servlets

DataArt

- Раньше сервлеты обычно описывали через специальный файл – web.xml
- Сейчас в большинстве случаев (но не всегда) можно обойтись без него и сконфигурировать приложение аннотациями
- Севрлеты – controller layer. Давайте добавим view layer (технология jsp)

Структура проекта

DataArt



Изменяем код сервлета

DataArt

```java
package controllers;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;

@WebServlet("/sayHello")
public class HelloWorldController extends HttpServlet {

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        String username = req.getParameter("userName");
        req.setAttribute("helloMessage", "hello, " + (username == null ? "anonymous" : username));
        req.getRequestDispatcher("/helloPage.jsp").forward(req, resp);
    }

}
```

Рассмотрим, что такое JSP и как вписать Java-код в Jsp

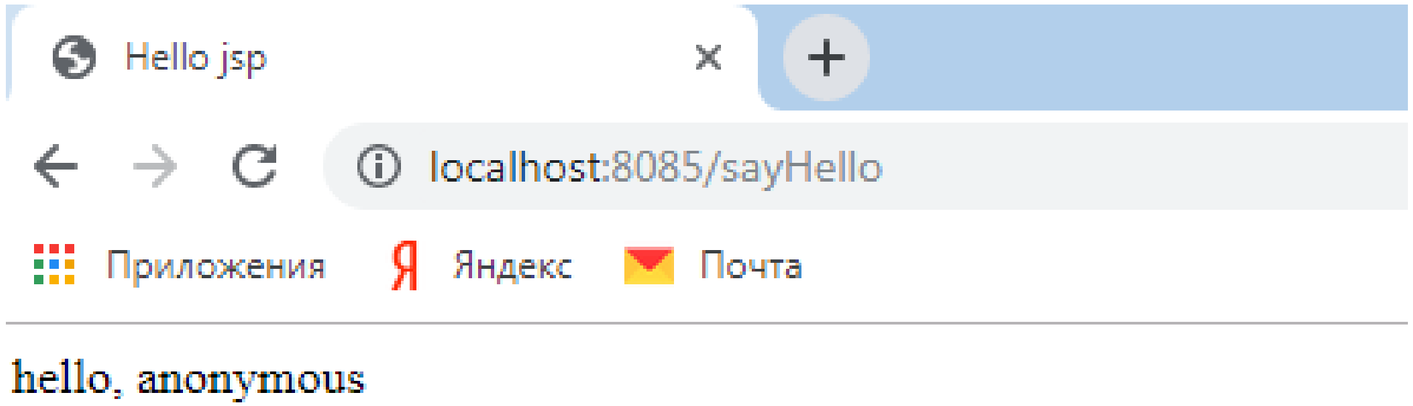
DataArt

servletexample | HelloWorldController.java | helloPage.jsp

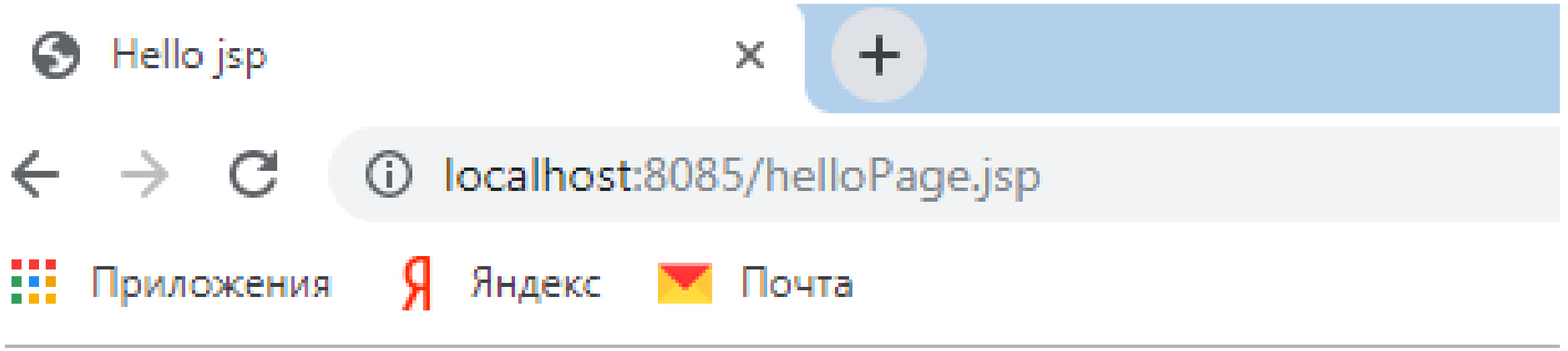
```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
<head>
    <title>Hello jsp</title>
</head>
<body>
<%
    Object helloMessage = request.getAttribute("helloMessage");
    if (helloMessage != null) {
%>

<%=helloMessage%>

<%
    }
%>
</body>
</html>
```

Пример работы

DataArt



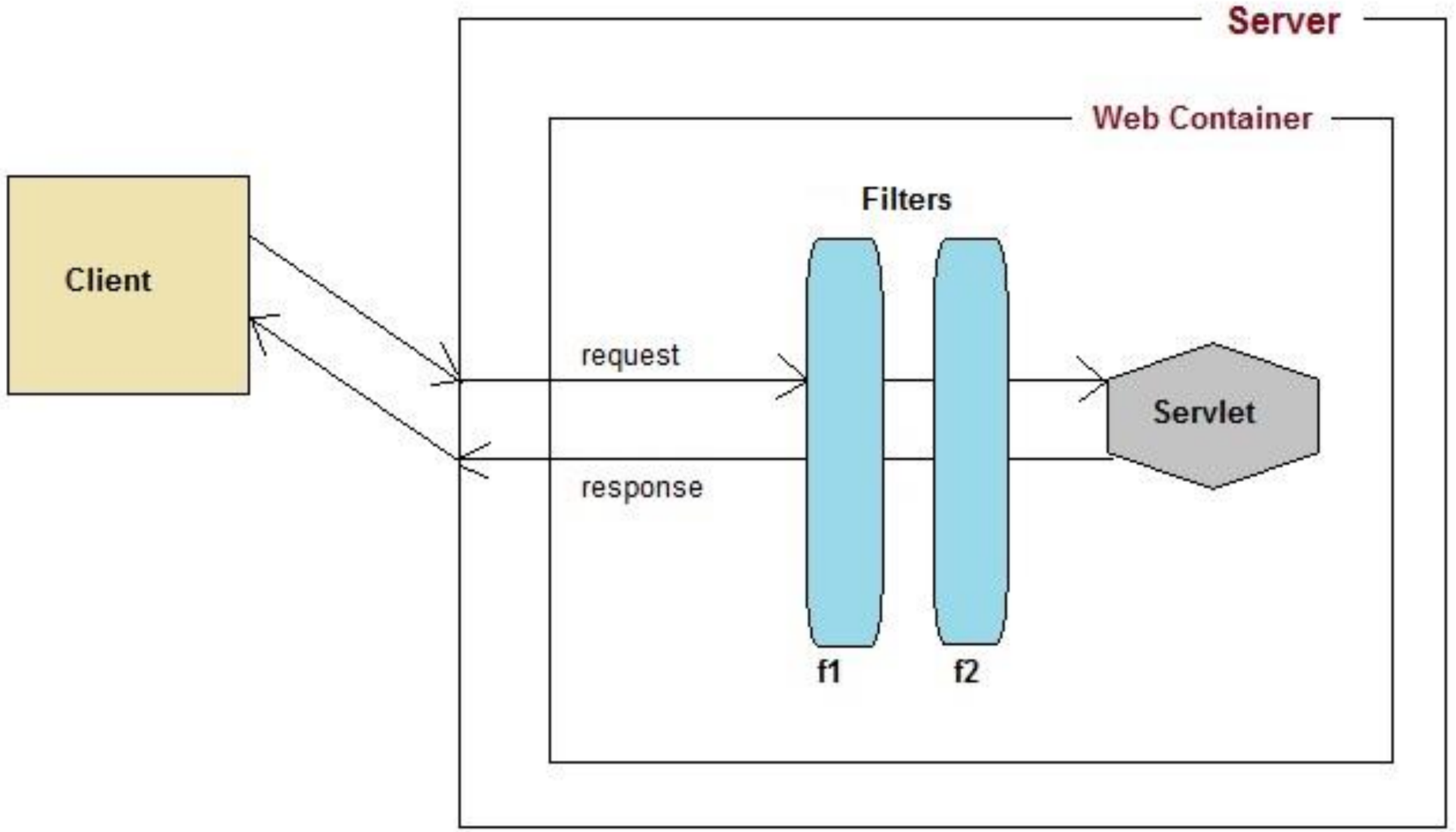
Проблема, у нас есть прямой доступ до view



Хотя мы и получили пустую страницу, мы видим в title, что jsp попыталась открыться

Закроем доступ с помощью WebFilter

DataArt

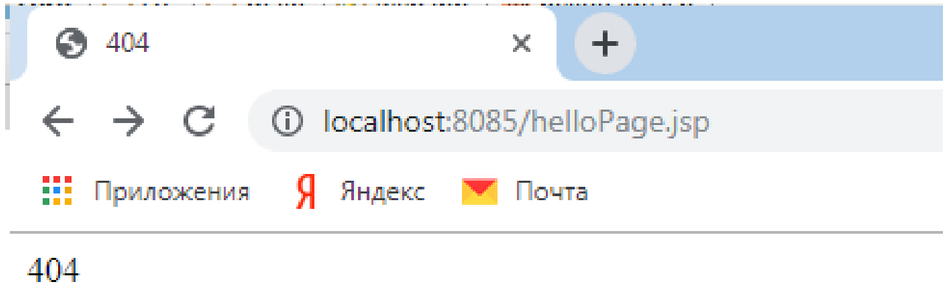


Закроем доступ с помощью WebFilter

DataArt

```
import java.io.IOException;

@WebFilter(urlPatterns = "*.jsp")
public class JspFilter implements Filter {


    @Override
    public void init(FilterConfig filterConfig) throws ServletException {}

    @Override
    public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain) throws IOException, Serv
        request.getRequestDispatcher("/notfound.jsp").forward(request, response);
        //chain.doFilter(request, response) это строка пропустила бы запрос сквозь фильтр
    }

    @Override
    public void destroy() {}
}
```

Закроем доступ с помощью WebFilter



vletexample | HelloWorldController.java | helloPage.jsp | JspFilter.java | notfound.jsp

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
<head>
    <title>404</title>
</head>
<body>
404
</body>
</html>
```

DataArt

Вопросы и ответы

Что будем делать

DataArt

- TODO – list, записная книжка дел, которые вам надо сделать.
- Сегодня просто заставим приложение вычитывать данные из файла и отображать их на HTML-странице.


DataArt