

УДК 004.4: 004.9: 528.9: 912.43

UDC 004.4: 004.9: 528.9: 912.43

РЕАЛИЗАЦИЯ СЕРВИСА ХРАНЕНИЯ ДАННЫХ¹**IMPLEMENTATION OF STORAGE DATA SERVICE**

Суханов Владимир Иванович
д.т.н., доцент

Sukhanov Vladimir Ivanovich
Dr.Sci.Tech., associate professor

*Уральский федеральный университет,
Екатеринбург, Россия*

Ural Federal University, Yekaterinburg, Russia

Описана программная реализация федеративного хранилища данных

In the article we present the program implementation of the federation data storage

Ключевые слова: БАЗА ДАННЫХ, ВЕБ-ПРИЛОЖЕНИЕ, СЕРВЕР

Keywords: DATA BASE, WEB-APPLICATION, SERVER

Введение

Одним из ключевых направлений развития информационных технологий являются облачные вычисления с возможностью доступа к данным, предоставляемым различными провайдерами на свободной основе. Такими данными являются публичные сведения о юридических лицах, включая реквизиты банков, справочники адресов, видов деятельности, нормативные документы и другие сведения. Эксперты Gartner считают, что стремление пользователей делиться своими данными и иметь к ним доступ с разных цифровых устройств приведет к тому, что к 2016 году треть всех данных будет храниться в «облаках» [1].

Материал статьи продолжает обсуждение проблемы реализации экспериментального образца программного обеспечения (ЭО ПО) сервиса федеративного хранилища для общественных данных, приведенной в [2]. Инструментальные средства реализации ЭО ПО обсуждались в работе [3]. Далее рассматриваются детали организации мониторинга и взаимодействия прикладных программ с хранилищем через веб-сервисы по протоколам SOAP и REST.

¹ Работа поддерживается Министерством образования и науки Российской Федерации, ГК №14.514.11.4014

Концепции сервиса хранения данных

Представлением данных на уровне хранения в репозитории и передачи пользователю является XML-документ. Следовательно, описание структуры хранимых данных следует выполнить в формате XML Schema (http://ru.wikipedia.org/wiki/XML_Schema). В этом случае схемы представления информации в XML файле становится некоторым аналогом описания таблицы реляционной СУБД или класса объектов в объектно-ориентированных базах данных. Языком манипулирования данными является XQuery, позволяющий исполнять запросы на выборку и форматирование информации к XML-документам и таблицам баз данных реляционных СУБД.

Общая схема хранилища показана на рисунке 1. Учет всех предоставляемых сервисом ресурсов ведется в репозитории, выполненном в СУБД PostgreSQL. Для обслуживания репозитория имеется веб-приложение для просмотра и редактирования учетной информации, выполняемого администратором системы.

Веб-интерфейс пользователей для загрузки ресурсов, получения копий ресурсов на локальные ЭВМ и выполнения запросов на извлечение данных из хранимых документов реализован как веб-приложение с авторизованным доступом и передачей данных по защищенному протоколу https. Доступ к ресурсам хранилища сторонних приложений выполняется при помощи вер-сервисов по протоколам SOAP и REST.

Все обращения к сервису хранения регистрируются в журнале операций центрального репозитория, просматриваются и анализируются администратором через веб-интерфейс подсистемы мониторинга.

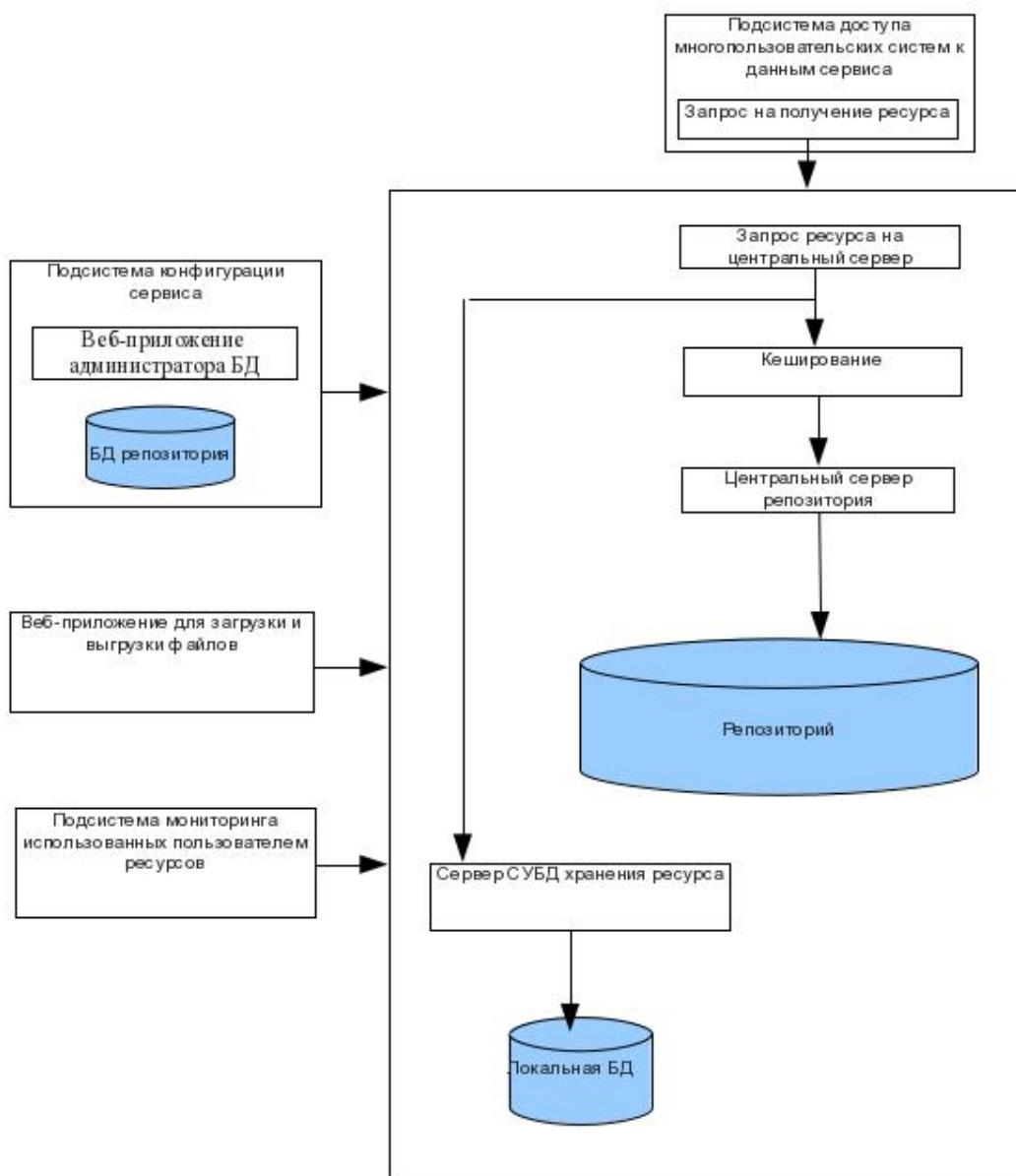


Рисунок 1 – Общая схема взаимодействия компонент сервиса

Разработка программы администрирования БД

Репозиторий системы является базой данных, в которой хранятся сведения о всей доступной информации в системе. По каждому ресурсу в репозитории имеются метаданные, тип и локализация источника, структуре таблицы или документа, формате хранения. Для разработки веб-приложения для администратора БД репозитория требуются следующие ресурсы:

- веб-браузер с поддержкой HTML 5;

- среда разработки приложений Netbeans 7.3.

Разработка включает следующие шаги.

- 1) В Netbeans создается новое веб-приложение с именем «adminrepo».
- 2) В Netbeans создаем сущности на основе таблиц базы данных. При этом создаем новое соединение с БД PostgreSQL, указывая адреса, имена, логины, пароли и другие параметры созданной ранее БД репозитория. Выбираем все созданные ранее таблицы. Размещаем классы в пакете «entities»
- 3) В Netbeans создаем JSF страницы на основе сущностей. Выбираем все созданные ранее сущности. Размещаем сгенерированные программы в следующих пакетах (папках):
 - 1) *facades* – фасады созданных сущностей;
 - 2) *controlers* – контроллеры создаваемых страниц сущностей;
 - 3) *pages* – страницы xhtml для просмотра и редактирования содержимого таблиц.

После генерации кода классов и страниц вручную откорректировать следующие фрагменты:

- 1) В классах сущностей для более понятного восприятия имен объектов в списках выбора. заменить тело метода `toString` на

```
public String toString() {
    return <Ключевое поле записи> + ": " + <Имя
экземпляра>;
}
```
- 2) На страницах создания и редактирования сущностей для полей типа «text» заменить тип компонента ввода *inputText* на *inputTextarea* для редактирования длинных многострочных текстов.

- 3) Отредактировать названия полей в интерфейсах пользователя, заменив английские слова на слова русского языка в файле *Bundle.properties*.
- 2) Для проверки работоспособности приложения запустить его на исполнение из среды Netbeans или из браузера по адресу <http://localhost:8080/adminrepo/>.

Для защиты приложения администратора от несанкционированного доступа используется проверка полномочий пользователя. Для этого в дескрипторе развертывания приложения *web.xml* добавляются ограничения безопасности:

```
<security-constraint>
    <web-resource-collection>
        <web-resource-name>repotxt</web-resource-
name>
        <url-pattern>/*</url-pattern>
        <http-method>GET</http-method>
        <http-method>POST</http-method>
    </web-resource-collection>
    <auth-constraint>
        <role-name>admins</role-name>
    </auth-constraint>
</security-constraint>
<login-config>
    <auth-method>BASIC</auth-method>
    <realm-name>file</realm-name>
</login-config>
<security-role>
    <role-name>admins</role-name>
</security-role>
```

В дескрипторе *glassfish-web.xml*:

```
<security-role-mapping>
    <role-name>admins</role-name>
    <group-name>admins</group-name>
</security-role-mapping>
```

В консоли администратора сервера Glassfish нужно добавить пользователя с именем *admin* и его паролем в разделе *Configurations – server-config – Security – Realms – File – Manage user*.

Веб-сервисы позволяют связывать сервер и клиента поверх протокола HTTP. Веб-сервисы могут быть реализованы разными способами: **JAX-WS** веб-сервисы SOAP и **JAX-RS** веб-сервисы REST.

При разработке сервисов пересылки файлов используется специальная технология мультидоступа *multipart*. Для разработки демонстрационного REST-сервиса загрузки файлов на сервер в среде Netbeans нужно создать новое веб-приложение в категории Maven и дополнить его следующим текстом [4].

Файл index.jsp:

```
<form action="upload/file/upload" method="post"
enctype="multipart/form-data">
    <p> Выбран файл : <input type="file" name="file"
size="35" /> </p>
    <input type="submit" value="Загрузить файл" />
</form>
```

Файл UploadFileService.java:

```
@Path("/file")
public class UploadFileService {
    @POST
    @Path("/upload")
```

```
@Consumes(MediaType.MULTIPART_FORM_DATA)
public Response uploadFile(
    @FormDataParam("file") InputStream
uploadedInputStream,
    @FormDataParam("file") FormDataContentDisposition
fileDetail) {
    String uploadedFileLocation = "uploaded/"
        + fileDetail.getFileName();
    writeToFile(uploadedInputStream,
uploadedFileLocation);
    String output = "Файл сохранен в : " +
uploadedFileLocation;
    return Response.status(200).entity(output).build();
}
}
```

Файл pom.xml дополнить строками:

```
<repositories>
  <repository>
    <id>maven2-repository.java.net</id>
    <name>Java.net Repository for Maven</name>
    <url>http://download.java.net/maven/2/</url>
    <layout>default</layout>
  </repository>
</repositories>
<dependencies>
  <dependency>
    <groupId>javax</groupId>
    <artifactId>javaee-web-api</artifactId>
    <version>6.0</version>
```

```
    <scope>provided</scope>
</dependency>
<dependency>
    <groupId>com.sun.jersey</groupId>
    <artifactId>jersey-server</artifactId>
    <version>1.8</version>
</dependency>
<dependency>
    <groupId>com.sun.jersey.contribs</groupId>
    <artifactId>jersey-multipart</artifactId>
    <version>1.8</version>
</dependency>
</dependencies>
```

Файл web.xml дополнить строками:

```
<servlet>
    <servlet-name>ServletAdaptor</servlet-name>
    <servlet-class>
com.sun.jersey.spi.container.servlet.ServletContainer
</servlet-class>
    <init-param>
        <param-name>
com.sun.jersey.config.property.packages</param-name>
        <param-value>com.user.upload</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
</servlet>
<servlet>
    <servlet-name>Faces Servlet</servlet-name>
```



```
<servlet-class>
javax.faces.webapp.FacesServlet</servlet-class>
<load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
<servlet-name>ServletAdaptor</servlet-name>
<url-pattern>/upload/*</url-pattern>
</servlet-mapping>
```

Для решения задач, стоящих перед ЭО сервиса, реализован сервис для пользователей, выполняющий следующие действия:

Получение перечня ресурсов, зарегистрированных на сервисе:

```
@GET
@Path("/getResources/")
@Produces(MediaType.APPLICATION_JSON)
List<Resource> getResources(@QueryParam("description") String
description);
```

Пример данного запроса <http://localhost:8180/storage-service/ConsumerService/getResources?description=test>. Для следующих запросов вызов аналогичен.

Получение перечня ресурсов, на которые подписан пользователь:

```
@GET
@Path("/getSubscribedResources/")
@Produces(MediaType.APPLICATION_JSON)
Set<Resource> getSubscribedResources(@QueryParam("email")
String email);
```

Получение перечня ресурсов, которыми владеет пользователь:

```
@GET
@Path("/getOwnedResources/")
```

```
@Produces(MediaType.APPLICATION_JSON)
Set<Resource> getOwnedResources(@QueryParam("email") String
email);
```

Перечень запросов к ресурсу:

```
@GET
@Path("/getQueryForResource/")
@Produces(MediaType.APPLICATION_JSON)
List<Query> getQueryForResource(@QueryParam("resourceId")
Long resourceId);
```

Выполнение запросов к ресурсу. В запросе передаются параметры:

```
@GET
@Path("/executeQueryOnResource/")
@Produces(MediaType.TEXT_XML)
String executeQueryOnResource(@Context UriInfo info)
```

Заключение

Работы в области создания федеративных хранилищ в мировой практике находятся в начальном состоянии, что позволяет надеяться на получение в этом направлении полезных для практики результатов. Этому способствует готовность рынка свободных инструментальных средств разработки веб-приложений и веб-сервисов на языках программирования Java, Python, Ruby on Rails, широко используемых в облачных технологиях, и имеющийся широкий опыт использования этих инструментов для разработки веб-технологий для облаков.

Список литературы

1 Облачные вычисления (мировой рынок) [Электронный ресурс] // TAdviser. - М., 2012. - Режим доступа: <http://www.tadviser.ru/index.php/> Статья: Облачные_вычисления_мировой_рынок (дата обращения: 15.01.2013).

2 Суханов В.И. Облачный сервис хранения данных / В.И. Суханов, О.Л. Чагаева // Политематический сетевой электронный научный журнал Кубанского государственного аграрного университета (Научный журнал КубГАУ) [Электронный ресурс]. – Краснодар: КубГАУ, 2013. – №02(86). – Режим доступа: <http://ej.kubagro.ru/2013/02/pdf/50.pdf>

3 Суханов В.И. Обработка XML-документов в JAVA EE / В.И. Суханов // Политематический сетевой электронный научный журнал Кубанского государственного аграрного университета (Научный журнал КубГАУ) [Электронный ресурс]. – Краснодар: КубГАУ, 2013. – №02(86). – Режим доступа: <http://ej.kubagro.ru/2013/02/pdf/44.pdf>

4 File upload example in Jersey. [Электронный ресурс]. Режим доступа: <http://www.mkyong.com/webservices/jax-rs/file-upload-example-in-jersey/> (дата обращения: 27.12.2012).

References

1 Cloud computing (world market) [electronic resource] // TAdviser. - М., 2012. - Mode of access: <http://www.tadviser.ru/index.php/> Article: Облачные_вычисления_мировой_рынок (date accessed: 15.01.2013).

2 V. Sukhanov. Cloud storage service / V.I. Sukhanov, O.L. Chagaev // Polythematic power electronic scientific journal of the Kuban State Agrarian University (Journal KubGAU) [electronic resource]. - Krasnodar KubGAU, 2013. - № 02 (86). - Mode of access: <http://ej.kubagro.ru/2013/02/pdf/50.pdf>

3 V. Sukhanov. Processing of XML-documents in a JAVA EE / VI Sukhanov // Polythematic power electronic scientific journal of the Kuban State Agrarian University (Journal KubGAU) [electronic resource]. - Krasnodar KubGAU, 2013. - № 02 (86). - Mode of access: <http://ej.kubagro.ru/2013/02/pdf/44.pdf>

4 File upload example in Jersey. [Electronic resource]. Mode of access: <http://www.mkyong.com/webservices/jax-rs/file-upload-example-in-jersey/> (date accessed: 27.12.2012).

Работа поддерживается Министерством образования и науки Российской Федерации, ГК №14.514.11.4014