Оглавление

[Концепция веб-сервисов 2](#_Toc285134327)

[Введение 2](#_Toc285134328)

[Основные положения модели веб-сервисов 3](#_Toc285134329)

[Серверная часть 6](#_Toc285134330)

[Анализ вариантов использования 6](#_Toc285134331)

[Выбор технологии реализации 8](#_Toc285134332)

[XML RPC в PHP 9](#_Toc285134333)

[CMS Joomla 10](#_Toc285134334)

[Joomla API и XML RPC 11](#_Toc285134335)

[Реализация бизнес-логики на базе Joomla API 12](#_Toc285134336)

[Программные модули 12](#_Toc285134337)

[Поддержка MVC 13](#_Toc285134338)

[Работа с БД 14](#_Toc285134339)

[Отображение интерфейса пользователя 15](#_Toc285134340)

[JavaScript и AJAX 16](#_Toc285134341)

[Клиентская часть 17](#_Toc285134342)

[Анализ вариантов использования 17](#_Toc285134343)

[Выбор технологии реализации 18](#_Toc285134344)

[Обзор Adobe Lightroom 19](#_Toc285134345)

[Средства автоматизации Lightroom 20](#_Toc285134346)

[Реализация бизнес-логики на базе Lightroom API 21](#_Toc285134347)

[Программные модули 21](#_Toc285134348)

[Работа с массивами фотографий 22](#_Toc285134349)

[Реализация сервиса публикации 23](#_Toc285134350)

# Концепция веб-сервисов

## Введение

Сеть Интернет стала общепризнанным фактором деловой и общественной жизни. Широкая распространенность и возросшая пропускная способность создают условия, при которых выгодно решать многие задачи при помощи интернет-технологий.

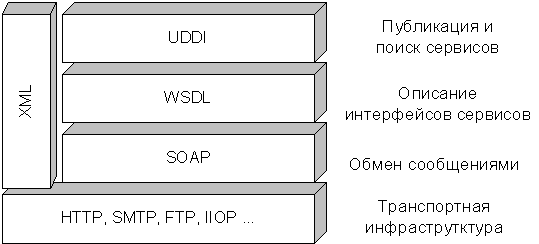
Однако Интернет объединяет в себе много различных платформ, а информация содержится в разнообразных источниках данных. Поэтому актуальна проблема связи таких разнородных данных, а также создания способа, который позволяет получать их в виде удобном для дальнейшей обработки.

Концепция веб-сервисов (Web Services) призвана решить эту задачу объединения, интеграции разнородных систем на основе открытых стандартов. Данная работа посвящена веб-сервисам, в ней кратко рассмотрены основные положения модели веб-сервисов, а также компоненты этой модели и технологии, используемые для их реализации. Практическая часть работы содержит небольшой пример, демонстрирующий разработку веб-сервиса и приложений использующих его. Пример основывается на реализации концепции веб-сервисов в рамках Java-технологий. Для понимания примера достаточно базовых знаний Java.

## Основные положения модели веб-сервисов

Веб-сервисы являются концепцией создания таких приложений, функции которых можно использовать при помощи стандартных протоколов Интернет. В настоящее время эту концепцию применяют и развивают многие ведущие компании в IT-области. Концепция веб-сервисов реализуется при помощи ряда технологий, которые стандартизованы World Wide Web Consortium (W3C).

Взаимосвязь этих технологий можно условно представить следующим образом.



Веб-сервисы являются одним из вариантов реализации компонентной архитектуры.

XML является фундаментом для создания большинства технологий, связанных с веб-сервисами.

Для удаленного взаимодействия с веб-сервисами используется Simple Object Access Protocol (SOAP). SOAP обеспечивает взаимодействие распределенных систем, независимо от объектной модели, операционной системы или языка программирования. Данные передаются в виде особых XML документов особого формата.

Согласно определению W3C, веб-сервисы это приложения, которые доступны по протоколам, которые являются стандартными для Интернет. Нет требования, чтобы веб-сервисы использовали какой-то определенный транспортный протокол. Спецификация SOAP определяет, каким образом связываются сообщения SOAP и транспортный протокол.

Наиболее часто реализуется передача SOAP сообщений по протоколу HTTP. Также широко распространено использование в качестве транспортного протокола SMTP, FTP, TCP.

Согласно определению W3C, "WSDL - формат XML для описания сетевых сервисов как набора конечных операций, работающих при помощи сообщений, содержащих документно-ориентированную или процедурно-ориентированную информацию". Документ WSDL полностью описывает интерфейс веб-сервиса с внешним миром. Он предоставляет информацию об услугах, которые можно получить, воспользовавшись методами сервиса, и способах обращения к этим методам.

Технология Universal Description, Discovery and Integration (UDDI) предполагает ведения реестра веб-сервисов. Подключившись к этому реестру, потребитель сможет найти веб-сервисы, которые наилучшим образом удовлетворяют его потребностям. Технология UDDI дает возможность поиска и публикации нужного сервиса, как человеком, так и программой-клиентом. Поиск и публикация в реестре предоставляется программе-клиенту как набор веб-сервисов реестра UDDI.

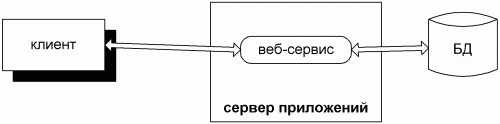
Веб-сервисы позиционируются как программное обеспечение промежуточного слоя. Использовать веб-сервисы могут как клиентские приложения, непосредственно работающие с пользователем, так и другие приложения (в том числе и другие веб-сервисы).

Веб-сервисы размещаются на серверах приложений.

Разработчики концепции веб-сервисов предлагают следующие сценарии применения веб-сервисов:

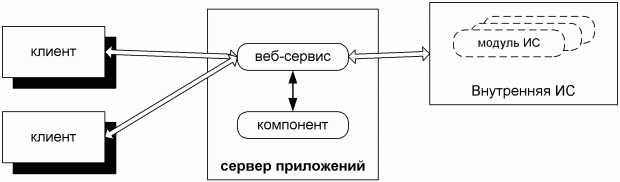
* Веб-сервисы как реализация логики приложения (бизнес-логики).

То есть, создание нового приложения бизнес-логика, которого реализуется в веб-сервисе



* Веб-сервисы как средство интеграции.

То есть, использование веб-сервиса как способа доступа удаленных клиентов к внутренней ИС компании, или для организации взаимодействия компонента (например, EJB, COM-компонента) с различными удаленными клиентами.



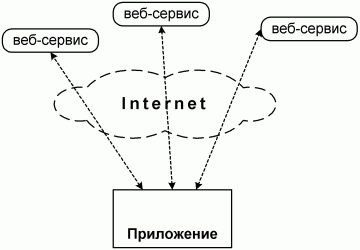
* Создание альтернативного сервиса.

В этом случае, при разработке нового веб-сервиса используется описание интерфейса уже существующего веб-сервиса. Таким образом, сервис имеет много потенциальных клиентов сразу с момента начала работы, а подключение к нему не требует существенных изменений на стороне клиента.

Как было сказано выше, концепция веб-сервисов включает в себя возможность ведения реестра веб-сервисов. Описание интерфейса может быть получено из такого реестра. После создания и внедрения нового веб-сервиса, имеет смысл зарегистрировать его в реестре. Тогда клиенты при поиске сервисов, реализующих исходный интерфейс, получат указание и на новый веб-сервис.

* Использование веб-сервиса как строительного блока при создании приложения.

Приложение может использовать веб-сервисы как удаленные компоненты, которые предоставляют определенную функциональность. Существуют различные сервисы, которые предоставляют качественное решение таких задач как аутентификация, ведение календаря, отправка сообщений, поиск, перевод и т. П

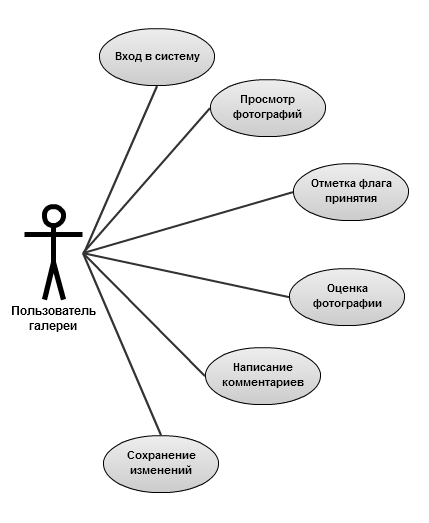


Кроме этого, и возможны другие варианты использования веб-сервисов. Например, существуют исследования по использованию веб-сервисов для построения распределенных вычислительных и информационных систем и одноранговых и со сложной иерархической структурой.

# Серверная часть

## Анализ вариантов использования

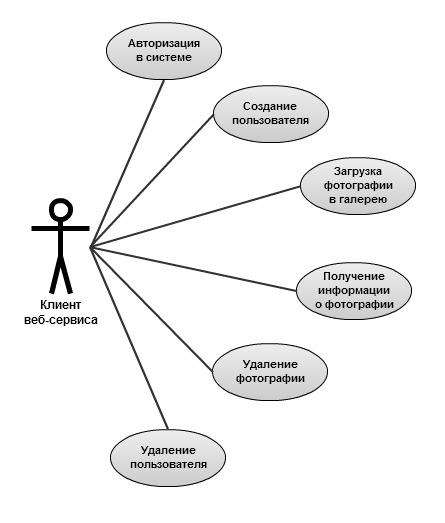
Серверная часть представляет собой объект для действий двух категорий объектов. Во-первых, это пользователи онлайн фотогалереи – именно она является фронтендом серверного программного обеспечения. Во-вторых, это клиентские системы, которые используют эту часть как веб-сервис. Исходя из этого, необходимо проанализировать две группы вариантов использования. На первой UseCase диаграмме изображены варианты использования системы пользователем фотогалереи.



Они включают в себя:

* вход в систему – пользователь проходит авторизацию и аутентификацию для определения списка фотографий, подлежащих просмотру и редактированию;
* просмотр фотографий – пользователь имеет возможность просмотра фотографий, опубликованных для него;
* отметка флага принятия – каждая фотография может быть принята / не принята пользователем, либо отметка может быть не проставлена. Этот признак позволяет фотографу отсеять ненужные снимки
* оценка фотографии – клиент проставляет оценку по 5-балльной шкале каждой фотографии. Это делается для того, чтобы выделить наиболее желаемые для дальнейшей обработки снимки;
* написание комментариев – здесь пользователь может оставить какие-либо пожелания для дальнейшей обработки фотографий;
* сохранение изменений – после того, как клиент проделал все необходимые манипуляции с опубликованными снимками, он сохраняет свои изменения, тем самым уведомляя фотографа об окончании процесса приёмки.

Вторая UseCase диаграмма иллюстрирует варианты использования серверной части системы в качестве веб-сервиса различными клиентами.



Она включает в себя:

* авторизация в системе – для совершения операций, клиентской системе необходимо авторизоваться при помощи пары логин-пароль;
* создание пользователя – в процессе публикации фотографий есть возможность создания нового пользователя фотогалереи с заданным именем и паролем;
* загрузка фотографии в галерею – публикация фотографий осуществляется по одной, при этом необходимо помимо самого графического файла указать пользователя, для которого выполняется публикация;
* получение информации о фотографии – запрос и получение флага принятия, рейтинга и комментариев для конкретной фотографии определенного пользователя;
* удаление фотографии – удаление опубликованной фотографии с сервера;
* удаление пользователя – удаление пользователя и всех его фотографий с сервера.

## Выбор технологии реализации

## XML RPC в PHP

## CMS Joomla

## Joomla API и XML RPC

## Реализация бизнес-логики на базе Joomla API

### Программные модули

### Поддержка MVC

### Работа с БД

### Отображение интерфейса пользователя

### JavaScript и AJAX

# Клиентская часть

## C:\Documents and Settings\Артем\Рабочий стол\Реализация\Пользователь lightroom.pngАнализ вариантов использования

## Выбор технологии реализации

## Обзор Adobe Lightroom

## Средства автоматизации Lightroom

## Реализация бизнес-логики на базе Lightroom API

### Программные модули

### Работа с массивами фотографий

### Реализация сервиса публикации