

УДК 681.3.07

Губский А.Н.

ОБЩАЯ АРХИТЕКТУРА СИСТЕМЫ «ЭЛЕКТРОННЫЙ КАМПУС»

Национальный технический университет Украины «Киевский политехнический институт», Киев, проспект Победы 37, 03056

UDC 681.3.07

Gubskiy A.N.

GENERAL ARCHITECTURE OF THE SYSTEM “ELECTRONIC CAMPUS”

*National Technical University of Ukraine “Kyiv Polytechnic Institute”,
Kiev, Pobedy Avenue 37, 03056*

В статье описывается многоуровневая архитектура построения сложной информационной системы в гетерогенной среде с возможностью применения различных технологий разработки.

Ключевые слова: кампус, гетерогенная среда, технологии разработки, многоуровневая архитектура.

In the article the multilevel architecture of creation of difficult information system in the heterogeneous environment with possibility of application of various technologies of development is described.

Keywords: campus, heterogeneous environment, technologies of development, multilevel architecture.

Введение. На сегодняшний день технологии разработки и подходы развиваются очень динамично. При построении первой версии системы «Электронный кампус КПИ» (ЭК КПИ) в разработке принимало участие большое количество специалистов различного уровня подготовки владеющих различными технологиями. Основной проблемой при этом была проблема координации и взаимодействия между различными командами.

Поэтому основной задачей при проектировании следующей версии системы являлась задача разработки такой архитектуры, которая позволила бы органично строить и развивать платформу в гетерогенной технологической среде.

Постановка задачи. Разработать архитектуру системы ЭК КПИ, которая позволила бы: организовать многослойную модульную систему, предусматривала бы возможность расширения системы новыми подсистемами без необходимости остановки всей системы, дала возможность удобной интеграции различных подсистема, обеспечила максимальное повторное использование реализованных компонент.

Описание архитектуры системы. В качестве стиля построения архитектуры был выбран REST-подход - стиль построения архитектуры распределенного приложения. Данные в REST передаются в виде небольшого количества стандартных форматов (в данном случае - XML, JSON). Сетевой протокол (как и HTTP) должен поддерживать кэширование, не должен зависеть от сетевого слоя, не должен сохранять информацию о состоянии между парами «запрос-ответ». Утверждается, что такой подход обеспечивает масштабируемость системы и позволяет ей эволюционировать с новыми требованиями.

Кроме того система разбита на три уровня, это сделано затем, что многоуровневая архитектура обеспечивает группировку связанной функциональности приложения в разных слоях, выстраиваемых вертикально, поверх друг друга. Функциональность каждого слоя объединена общей ролью или ответственностью. Слои слабо связаны, и между ними осуществляется явный обмен данными.

Правильное разделение приложения на слои помогает поддерживать строгое разделение функциональности, что в свою очередь, обеспечивает гибкость, а также удобство и простоту обслуживания. При строгом разделении на слои компоненты одного слоя могут взаимодействовать только с

компонентами того же слоя или компонентами слоя, расположенного прямо под данным слоем.

Основные уровни и подуровни системы ЭК КПИ (рис.1):

- Уровень СУБД.
- Уровень ядра системы.
 - Слой системы аутентификации.
 - Слой системы маршрутизации запросов.
- Уровень приложений окружения.

Реализация API информационной системы «Электронный кампус КПИ» (ИС ЭК КПИ). API информационной системы ЭК КПИ представляет собой специфицированный набор методов доступных по HTTP протоколу. Все методы реализовываются на стороне ядра системы и доступны клиентским приложениям после прохождения процедуры аутентификации и авторизации пользователя.

Стандартным форматом передачи данных являются JSON и XML. Для стандартизации функций API был разработан ряд соглашений:

1. Связь между компонентами приложений (в том числе между двумя серверными подсистемами или интерфейсом и сервером) осуществляется через вызовы сетевого API по протоколам HTTP и HTTPS.

2. Идентификация приложений и пользователей происходит через механизм cookie: после аутентификации вызывающая сторона получает SID, который должна предъявлять при всех последующих вызовах сетевого API (имя cookie фиксируем "SID", т.е. session identifier).

3. Сетевые вызовы методов API идентифицируются через URL каждого метода, а параметры передаются двумя способами: для метода HTTP GET - как параметры в URL, а для метода HTTP POST - как параметры в POST запросе.

4. Ответы сетевого API могут быть в двух форматах: JSON и XML. Метод с одной логикой может иметь несколько вариантов вызова с разными форматами ответа.

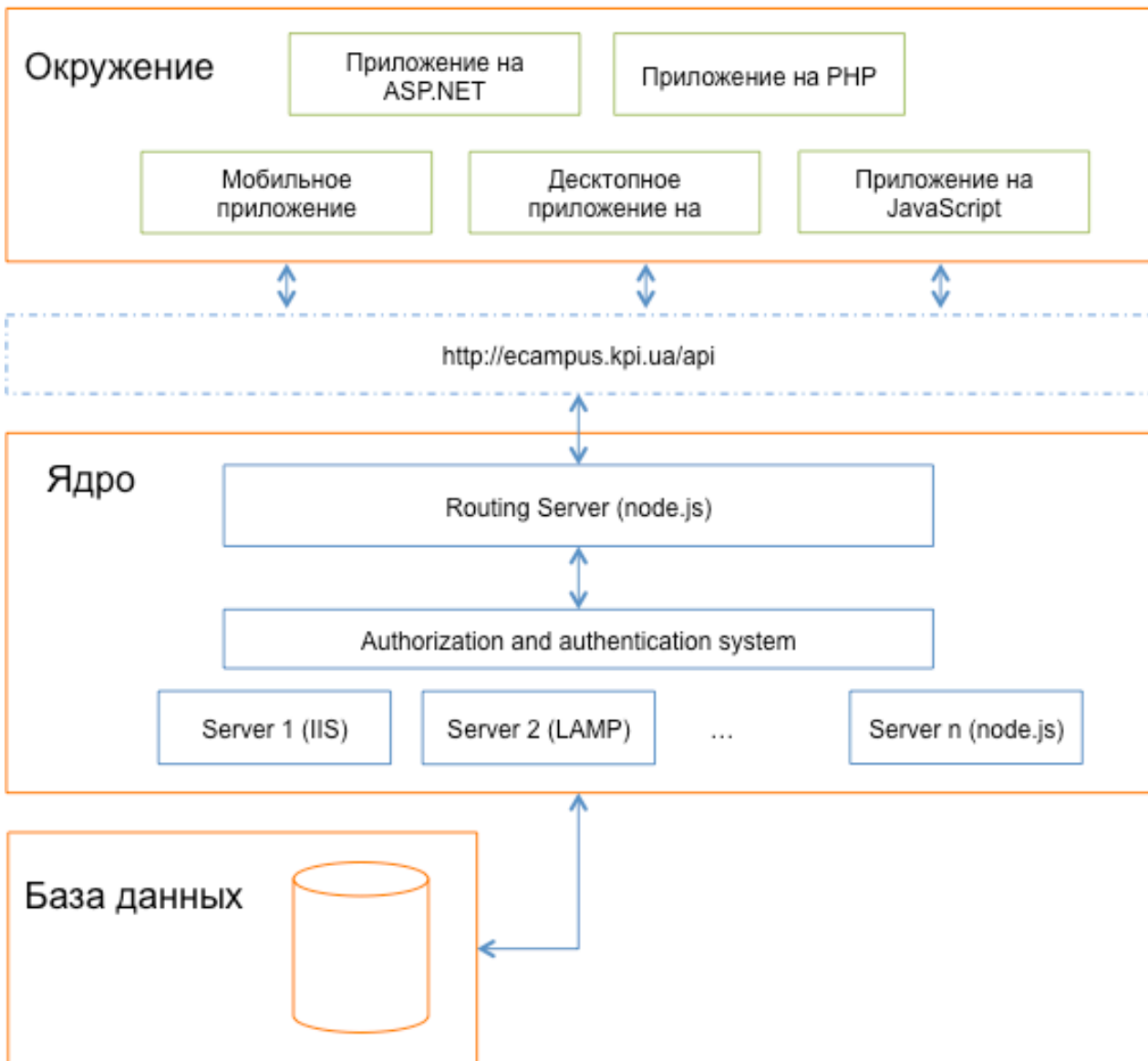


Рис. 1. Общая схема архитектуры информационной системы ЭК КПИ

Например:

1) выдача списка отфильтрованных адресатов в JSON (для веб приложения с последующей клиентской шаблонизацией или программного вызова из другого серверного компонента или мобильного приложения:

`http://ecampus.kpi.ua/api/messaging/FilterList/json?page=1&another_param=another_value`

2) выдача списка отфильтрованных адресатов в XML:

`http://ecampus.kpi.ua/api/messaging/FilterList/xml?page=1&another_param=another_value`

5. Формат ответа определяется "расширением", т.е. префиксом на конце URL: *.json - для JSON, *.xml - для XML. Ответ JSON имеет вид сериализованного объекта: {"Field1": "value1", "Field2": "value2", ...}.

6. В JSON ответах всегда присутствует поле "Result", которое имеет такие значения:

- Ok - выполнено успешно.
- Error – ошибка.
- Expired - сессионный cookie SID недействительна.

Система аутентификации и авторизации.

Система маршрутизации перенаправляет запросы, приходящие на `esampus.kpi.ua/api`, на соответствующие сервера.

Например, сервисы, отвечающие за выборку данных для статистики, находятся на сервере `server1.esampus.kpi.ua`, а сервисы, отвечающие за выборку информации о студентах и их принадлежности к учебным группам – находятся на `server2.esampus.kpi.ua`.

Маршрутизатор согласно таблице определяет какой запрос каким сервером должен обрабатываться (рис.2).

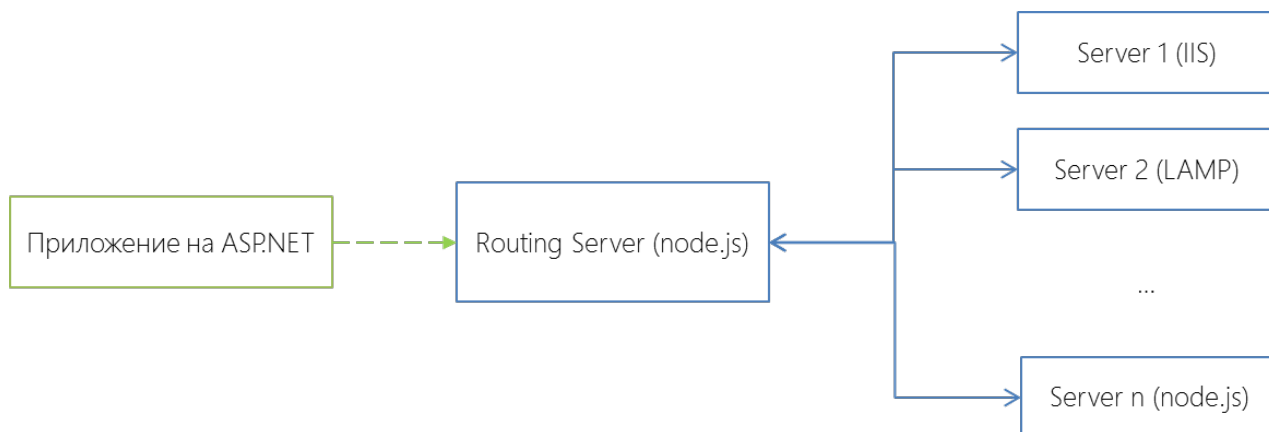


Рис. 2. Схема логики функционирования подсистемы маршрутизации информационной системы ЭК КПИ

Основные возможности маршрутизатора:

1. При маршрутизации ресурсов - быстрая отдача статических данных.

2. Исполнение JavaScript, уже в двух режимах из файла и кеша, так же, при изменении уже закешированных файлов система автоматически атуализирует кеш.

3. Маршрутизация запросов по регулярным выражениям на другие хосты и порты.

4. Серверная и клиентская шаблонизация с помощью одного и того же шаблонизатора.

5. Встроенный брокер трансляции сообщений на основе протокола SSE.

6. Поддержка сервера WebSockets.

7. Драйвера доступа к СУБД: mongodb, mysql, indexeddb, websql.

8. Поддержка кластерного режима, т.е. запускается несколько процессов Node.js и между ними обеспечивается соединение через IPC, главный процесс, принимает запросы от клиентов и передает дочерним процессам id сокетов для обработки.

9. Поддержка механизма "sticky session" т.е. связывание сессий пользователей к одному процессу, что исключает передачу на обработку запросов от этого пользователя на процесс, который его не обрабатывал и не имеет нужных развернутых в памяти структур или сессионных данных (пока только для процессов node.js).

Спецификация приложений окружения. Окружение представляет собой различные программы и сервисы, которые выступают в качестве клиентов API. В качестве клиентских приложений могут выступать:

- веб-приложения;
- мобильные приложения;
- десктопные приложения.

Поскольку разработкой различных модулей занимаются разные команды и программисты, для унификации внешнего вида модулей и элементов управления используется доработанная версия Twitter Bootstrap.

Выводы. Благодаря разработанной архитектуре система «Электронный кампус КПИ» может динамично развиваться при участии различных групп специалистов владеющих разными технологиями.

Каждый модуль системы может быть заменен модулем аналогичным по функционалу без необходимости внесения каких-бы то ни было изменений в архитектуру системы.