

УДК 004.738.5

ВЫБОР ВЕБ-СТЕКА ДЛЯ РЕАЛИЗАЦИИ ЦИФРОВОЙ СРЕДЫ ПРЕДОСТАВЛЕНИЯ ТРАНСПОРТНЫХ УСЛУГ

Давыдовский Михаил Альбинович,

канд. техн. наук, доцент, доцент кафедры вычислительных систем,
сетей и информационной безопасности,

e-mail: madavidovsky@gmail.com,

Институт управления и цифровых технологий,
Российский университет транспорта (МИИТ), г. Москва

Переход транспорта на цифровые технологии требует создания целого ряда сервисов обеспечения перевозок. Их разработка ведется на основе современных веб-технологий. В данной статье рассматриваются основные стеки, используемые при разработке веб-приложений. Описаны компоненты веб-стеков, предназначенные для обработки данных на различных уровнях реализации приложений. Проводится анализ особенностей веб-стеков. Рассмотрены синхронный и асинхронный режимы обработки информации, производительность программ обработки информации на стороне сервера, используемые системы управления реляционными и NoSQL базами данных, серверные языки программирования. Описаны преимущества веб-стеков при разработке определенного вида приложений. Основное внимание уделено сравнению стеков, использующих язык PHP, и стеков на основе программной платформы Node. В результате анализа сделан вывод о преимуществе стека, использующего платформу Node, для реализации интегральной цифровой среды предоставления транспортных услуг.

Ключевые слова: цифровые технологии на транспорте, веб-технологии, веб-стеки, базы данных, серверные языки программирования, PHP, Node

SELECTING A WEB STACK FOR IMPLEMENTATION OF A DIGITAL ENVIRONMENT FOR THE PROVISION OF TRANSPORT SERVICES

Davydovskiy M.A.,

candidate of technical sciences, associate professor, associate professor department of computing systems,
networks and information security,

e-mail: madavidovsky@gmail.com,

Institute of Management and Digital Technology,
Russian University of Transport (MIIT), Moscow

The transition of transport to digital technology requires the creation of a number of transportation services. Their development is based on modern web technologies. This article discusses the main stacks used in developing web applications. The components of web stacks designed to process data at various levels of application implementation are described. The analysis of features of web stacks is carried out. The synchronous and asynchronous information processing modes, the performance of server-side information processing programs, the relational and NoSQL database management systems used, and server programming languages are considered. The advantages of web stacks for developing a certain type of application are described. The focus is on comparing stacks using the PHP language and stacks based on the Node software platform. As a result of the analysis, a conclusion was made about the advantage of the stack using the Node platform for implementing an integrated digital environment for providing transport services.

Keywords: digital technologies in transport, web technologies, web stacks, databases, server programming languages, PHP, Node

DOI 10.21777/2500-2112-2019-4-34-41

Введение

Использование цифровых технологий на транспорте является в настоящий момент приоритетным направлением его развития. Интегральная цифровая среда предоставления транспортных услуг разрабатывается с целью обеспечения пользователей услугами компаний, перевозящих грузы, и компаний, выполняющих сопутствующие услуги, например, хранение грузов. Данная информационная система разрабатывается в рамках проекта создания «цифровой железной дороги» [2; 5] в ОАО «РЖД».

Для реализации такого рода систем используется клиент-серверная архитектура обработки данных и веб-технологии. Клиент и сервер – это логические процессы, выполняющие обработку данных на удаленных друг от друга вычислительных устройствах. Основные функции системы реализуются на сервере, а доступ к ним осуществляется пользователями, как правило, с помощью веб-приложений либо приложений, установленных на мобильные устройства.

Для работы сайта или приложения, доступного через интернет и работающего в браузере, необходим целый набор программ или программных пакетов, которые последовательно обрабатывают поток данных, циркулирующий между клиентом и сервером. Такой набор программ называют веб-стеком. Веб-стек включает ряд компонентов. Это операционная система, установленная на сервере, веб-сервер, система управления базами данных, серверный язык программирования. В стек не включается сам браузер (Google Chrome, FireFox, Internet Explorer, Safari и др.) и операционная система клиента, в которой работает браузер (Windows, Mac OS, Linux).

Существует достаточно большое количество стеков, используемых при разработке веб-приложений. Каждый из них имеет свои особенности и преимущества при реализации определенного класса приложений.

Компоненты веб-стека

Операционная система, обычно это Windows или Linux, используется на сервере для выполнения всех программ.

Веб-сервер, обычно это Apache или nginx, обрабатывает запросы, посылаемые клиентом (браузером), на предоставление некоторой информации (контента). Этим контентом могут быть статические страницы, написанные на языке HTML, изображения, видео- и аудиоинформация. Контент может также формироваться динамически на основе информации, которая хранится в базе данных. Для формирования динамического контента используются серверные языки программирования. Динамический контент передается клиенту, либо в виде страниц на языке HTML, либо как некоторый набор данных в специальном формате, обычно JSON.

Система управления базами данных, обычно MySQL или MariaDB, используется для создания базы данных и поиска информации по запросам клиента. Применяются и другие системы управления базами данных, например, PostgreSQL, Microsoft SQL Server, Oracle, BigTable, MongoDB, Cassandra.

Серверный язык программирования используется для написания программ на стороне сервера, которые обрабатывают запросы клиентов и формируют динамические страницы на основе информации, содержащейся в базе данных. Имеется большое количество скриптовых языков, например, PHP, JavaScript, Python, Ruby, Perl. Также могут быть использованы и компилируемые языки, например C++, C#, Java, Haskell.

Веб-стеки

LAMP стек [3; 7] использует операционную систему Linux (L), веб-сервер Apache (A), базу данных MySQL (M) и язык PHP (P) [10]. В настоящее время это самый популярный стек для разработки динамических сайтов и веб-приложений. Имеются следующие модификации этого стека:

- LEMP стек вместо Apache использует nginx;
- LLMP стек вместо Apache использует lighttpd. Lighttpd – это быстрый защищенный веб-сервер, который работает как в операционной системе Linux, так и Microsoft Windows;

- LAPP стек вместо MySQL использует PostgreSQL;
- WAMP стек вместо Linux использует операционную систему Microsoft Windows;
- XAMPP стек может работать как в Linux, так и в Windows и использует язык PHP или Perl.

WISA стек использует только продукты фирмы Microsoft:

- операционную систему Windows;
- веб-сервер Internet Information Services (IIS);
- систему управления базами данных Microsoft SQL Server;
- платформу для разработки веб-приложений ASP.NET.

Платформа ASP.NET позволяет писать код приложения на разных языках, входящих в комплект .NET Framework. Это языки: C#, VisualBasic.NET, J#, JScript.NET и др. Эта платформа использует технологию AJAX и шаблон проектирования MVC (Model-View-Controller), который позволяет разделить модель данных, внешнее представление и управление данными.

MAMP стек – это стек, работающий в операционной системе macOS фирмы Apple. Этот стек включает веб-сервера Apache и nginx, систему управления базами данных MySQL, языки программирования PHP, Perl и Python. Данный стек позволяет работать с системой управления контентом WordPress.

Серверная платформа Node [9] использует скриптовый язык программирования JavaScript. Она не требует использования Apache или другого веб-сервера, т.к. выполняет его функции. Поэтому в названии стеков с этой платформой не указывается веб-сервер, а на первом месте стоит система управления базами данных.

MEAN стек [7; 8] использует следующие программы:

- MongoDB (M) – NoSQL-система управления базами данных, хранящая данные в формате «ключ-значение»;
- Express (E) – фреймворк для разработки серверной части веб-приложений на платформе Node [1]. Он является стандартным каркасом для разработки приложений на Node и содержит ряд методов, упрощающих код на Node;
- AngularJS (A) – фреймворк для разработки клиентской части одностраничных веб-приложений на платформе Node. Он включает функции обработки данных JSON-формата, использует MVC-шаблон проектирования веб-приложения, позволяет разработать веб-приложение, которое переносит часть нагрузки по обработке данных со стороны сервера на сторону клиента;
- Node (N) – платформа разработки серверных приложений.

Модификациями стека MEAN являются:

- MERN стек, в котором AngularJS заменен на React. React – библиотека на JavaScript для разработки пользовательских интерфейсов одностраничных и мобильных приложений;
- MEEN стек, в котором AngularJS заменен на Ember.js. Ember.js – это фреймворк, предоставляющий каркас для разработки веб-приложений с использованием MVC-шаблона.

JAMstack включает в себя такие компоненты, как JavaScript (J), API (A) и Markup (M). Это новая технология подготовки статических сайтов. JavaScript обеспечивает выполнение запросов пользователя, обрабатываемых на стороне клиента. Для реализации интерфейса пользователя может использоваться один из таких фреймворков, как Vue.js, React или Angular. Обработка на стороне сервера запросов клиента, формируемых с помощью программы на языке JavaScript, и все операции по взаимодействию с базой данных выполняются с помощью некоторого программного интерфейса (API). API может быть разработан применительно к конкретному сайту или, что наиболее часто, используются API, реализуемые сервисами, которые предоставляют сторонние компании.

Markup – это термин, означающий, что страницы сайта должны быть предварительно сформированы перед его развертыванием в сети. Это обычно делается с помощью генераторов статических сайтов (SSG, Static Site Generator). Например, можно воспользоваться генераторами Gatsby, Jekyll, Hugo, Nexo или Nuxt. Также могут быть использованы инструменты разработки веб-приложений на стороне клиента (Frontend JavaScript module bundlers). Например, Parcel, Webpack, FuseBox, Browserify. Эти инструменты формируют страницу на стороне клиента с помощью программ на JavaScript.

В стандартном процессе доступа к странице сайта пользователь указывает адрес страницы и может передать также некоторую информацию, используемую для формирования запроса. Сервер обраба-

тывает запрос клиента и на основе шаблона страницы и данных из базы формирует страницу с помощью программ, написанных, например, на языке PHP или JavaScript, а затем отправляет ее клиенту. Для выполнения указанных действий требуются хостинг, т.е. услуга по размещению на сервере ресурсов (шаблонов, кода, базы данных), необходимых для работы динамических сайтов. При использовании JAMstack нет необходимости в таком хостинге. Страницы заранее подготовлены, поэтому не требуется каждый раз при обращении к странице выполнять ее формирование. Сами страницы выкладываются в GitHub или на сервера для хранения файлов CDN. Однако, технология JAMstack не годится для случаев, когда требуется частое обновление содержимого страниц или формирование новых страниц, обновление информации в режиме реального времени, реализация интерактивных функций. Поэтому данная технология не может быть использована для реализации обработки данных в интегральной цифровой среде.

Серверные технологии Node и PHP

Node и PHP являются в настоящее время наиболее широко используемыми технологиями разработки серверной части веб-приложений.

PHP – это язык программирования для создания динамических сайтов и веб-приложений. Помимо языка PHP для реализации серверной части потребуется также веб-сервер, принимающий запросы клиентов, поступающие через интернет, и посылающий им ответы. Таким веб-сервером обычно является Apache.

Node – это серверная платформа для написания программ на языке программирования JavaScript. Node одновременно является и веб-сервером, поэтому для него не нужен Apache.

PHP создал Расмус Лердорф (Rasmus Lerdorf) в 1995 году. PHP – это скриптовый язык с открытым исходным кодом для написания серверной части веб-приложений. Скрипты на PHP могут включать в себя не только код на PHP, но также фрагменты кода на CSS (таблицы стилей), HTML (язык гипертекстовой разметки), программы на JavaScript, которые выполняются браузером на клиентской стороне и даже просто текст. Этот язык используется в таких проектах, как Facebook, Wikipedia, Yahoo, WordPress.

Node разработал Райан Даль (Ryan Dahl) в 2009 году. С появлением Node стала возможной разработка серверной части веб-приложений на языке JavaScript. Node – это приложение с открытым исходным кодом. Node является асинхронной управляемой событиями платформой с неблокирующими операциями ввода-вывода. Node позволяет разрабатывать быстрые и масштабируемые приложения. Он используется такими компаниями, как Microsoft, Uber, Netflix, Yahoo, LinkedIn и PayPal.

Node основан на ядре JavaScript V8, разработанном для Google Chrome. К отличительным особенностям V8 относится прямая компиляция в машинный код и средства оптимизации, обеспечивающие высокую скорость работы Node.

Сравнение приложений

PHP и Node имеют каждый свои преимущества [6]. PHP предпочтительнее Node при разработке приложений, которые не требуют частого взаимодействия между клиентом и сервером. Наилучшими вариантами приложений с PHP являются:

- приложения, использующие стек программных средств LAMP;
- системы управления контентом (например, Drupal и WordPress написаны на PHP);
- приложения, которые потребляют много процессорного времени. Например, научные приложения и программы прогноза погоды;
- блоги и коммерческие приложения, для построения которых используются CMS;
- финансовые приложения.

Node имеет преимущество при разработке следующих приложений:

- приложения, использующие современную технологию Single Page Applications. Например, индивидуальные сайты;

- приложения, управляемые данными (data-driven applications);
- приложения, которые должны хорошо масштабироваться к увеличению нагрузки на приложения из-за поступления большого количества запросов от клиентов. Преимущество достигается за счет использования неблокирующих операций ввода-вывода и технологии обработки событий;
- приложения, которые должны работать в реальном масштабе времени, например, чаты, системы потокового видео, навигационные системы;
- приложения, использующие такие технологии, как Angular и React (программные стеки – MEAN и MERN).

Способ работы программы и скорость выполнения

Существует два способа работы программы: синхронный и асинхронный. В синхронном варианте программа выполняется последовательно, один оператор за другим оператором кода. Следующий оператор выполняется только по завершению выполнения предыдущего оператора. В асинхронном варианте несколько операторов программы могут выполняться в одно и то же время.

Программа на PHP выполняется в синхронном режиме. Только некоторые приложения используют асинхронную технику. Если какая-то функция требует большого ресурса времени на выполнение, то следующий за ней оператор будет ждать ее выполнения. В результате клиент может долго ждать ответа на свой запрос.

Node работает в асинхронном режиме. Операции выполняются параллельно. Нет необходимости ожидания выполнения функций. Поэтому программа на Node работает быстрее, чем программа на PHP.

Код на PHP выглядит проще, т.к. не требует понимания таких понятий Node, как замыкания, callback функции, управление событиями.

Производительность

Язык PHP и язык JavaScript, который используется в Node, являются интерпретируемыми языками.

При обращении клиента к серверу, написанном на PHP, запрос поступает сначала на веб-сервер, например, Apache. Он запускает интерпретатор PHP, который выполняет код на PHP.

Node не требует использования веб-сервера. Хотя обычно Node работает совместно с фреймворком Express, но последний имеет небольшой набор быстро выполняемых функций.

Интерпретатор Node работает быстрее интерпретатора PHP, т.к. использует технологию Google Chrome V8. В большинстве тестов, представленных в [11], Node показывает лучшие результаты, чем PHP.

Для обмена информацией между клиентом и сервером часто используется формат данных JSON. В PHP для работы с этим форматом данных используются функции `json_encode` и `json_decode`. В Node используются аналогичные функции `JSON.stringify` и `JSON.parse`, которые работают быстрее.

Программа на PHP генерирует на сервере страницу на языке HTML и отправляет ее клиенту. На стороне клиента браузеру проще работать с чистым HTML кодом без использования JavaScript. С другой стороны, использование технологии AJAX позволяет формировать страницу на стороне клиента и пересылать только данные, а не всю страницу на HTML. Такая технология используется при разработке сайта с использованием Node. Данная технология требует пересылки меньшего объема данных между клиентом и сервером и поэтому работает быстрее.

PHP использует стандартную клиент-серверную модель, в которой каждый запрос клиента на сервер инициирует новое приложение, устанавливает соединение с базой данных, извлекает из базы информацию для формирования HTML-кода. В Node нет необходимости инициализации приложения при каждом запросе. Можно использовать, например, один и тот же объект, работающий с базой данных, для разных запросов.

PHP использует модель блокировок. При запросе к базе данных необходимо дождаться ответа, чтобы продолжить выполнение следующего кода программы. В Node нет необходимости ждать ответа.

Можно создать функцию обратного вызова (callback), которая прослушивает процесс выполнения доступа к базе данных и выполнится после завершения этого процесса.

PHP использует веб-сервер, который работает в многопоточном режиме. Для каждого соединения клиент-сервер создается один поток. Node работает в однопоточном режиме, что упрощает взаимодействие, однако требует разработки асинхронных приложений, что несколько труднее для программиста. В некоторых случаях возможно использовать веб-сервер и для работы с Node.

Работа с базами данных

В большинстве случаев программа на PHP работает с реляционными базами данных. Это такие системы управления базами данных, как MySQL, MariaDB, Microsoft SQL Server, PostgreSQL, Oracle. Хотя программа на PHP может работать и с NoSQL базами данных [4], но на практике такие приложения встречаются редко.

Node хорошо работает как с реляционными базами данных, так и с NoSQL базами данных. При этом предпочтение отдается базам данных NoSQL. Например, MongoDB, CouchDB, Apache Cassandra, Apache Ignite, YugabyteDB. Для работы с графами используются системы Neo4j, JanusGraph.

Стеки разработки

В случае необходимости использовать разные приложения, написанные на разных языках программирования, и осуществлять взаимодействие между этими приложениями потребуются значительные усилия по написанию эффективного кода программы.

Программа на языке PHP вынуждена взаимодействовать с программами, написанными на разных языках. В большинстве случаев программа на PHP является частью стека LAMP (Linux, Apache Server, MySQL, PHP).

Node обычно использует стек MEAN (MongoDB, Express, Angular, Node) или стек MERN (MongoDB, Express, React, Node). Для написания кода программы необходим только один язык JavaScript.

Стек LAMP хорошо подходит для разработки приложения, работающего на одном сервере. Если возникает необходимость масштабирования приложения на несколько серверов, то преимущество имеют стеки MEAN и MERN.

Фреймворки

Node использует различные фреймворки для разработки приложений. Это Derby, Express, Meteor, Sails и другие.

PHP также использует фреймворки для облегчения и быстрой разработки веб-приложений. Они помогают создавать приложения на основе Agile технологии разработки программного обеспечения, создавать защищенные веб-приложения. Это такие фреймворки, как Laravel, Codeigniter, Symphony, CakePHP, FuelPHP, Phalcon.

Для быстрой разработки приложений на PHP используются системы управления контентом (CMS – Content Management System) WordPress, Joomla, Drupal и другие.

Стиль программирования

Сложно оценивать, на каком языке удобнее писать программы. Технология Node появилась позже, чем PHP, и ее использование только набирает обороты. Язык JavaScript по отношению к PHP выглядит более современным. В нем учтены недостатки, найденные в PHP. Например, в силу долгой истории, в PHP можно использовать несколько разных функций для выполнения одних и тех же действий. Этого нет в Node.

В PHP используется техника смешивания кода, когда в код на PHP легко могут быть встроены, например, части кода на HTML. Такая практика позволяет быстро писать код, но со временем код становится запутанным.

В Node и PHP используется современная технология разработки MVC.

Надо отметить, что PHP постоянно развивается. В версию PHP 8 предполагается встроить JIT-компилятор, позволяющий получать код, выполняемый в виртуальной машине Zend VM, что увеличит скорость выполнения программ на PHP.

Node также имеет преимущество в том, что как на клиенте, так и на сервере используется один и тот же язык программирования JavaScript.

Использование в существующих проектах

Согласно данным W3Techs [12], в настоящее время для разработки серверной части сайтов PHP используется в подавляющем большинстве проектов (79 %). JavaScript же используется лишь в 0,7 % проектов. Это определяется тем, что Node более молодая технология.

Примером успешного применения Node в проектах может служить платформа потокового видео Netflix. Переход от первоначального варианта проекта, сделанного на Java, на платформу Node, улучшил время загрузки Netflix на 70 %. Это время между обращением клиента к приложению Netflix и моментом, когда пользователь может начать с ним работать.

Сервис платежной системы PayPal был первоначально разработан на JSP (Java Server Pages). По мере развития популярности сервиса оплаты PayPal увеличивалось количество транзакций, которые нужно было успевать обрабатывать. В результате PayPal был переписан на Node.

Другим примером является сервис заказа такси Uber. Благодаря Node сервис позволяет выполнять 2 миллиона вызовов удаленных процедур в секунду.

Платформа Node используется также в таких крупных проектах, как торговая платформа eBay, крупнейший мировой регистратор GoDaddy, социальная бизнес-сеть LinkedIn, сеть розничной торговли Walmart, поисковик и почтовый сервер Yahoo, платформа электронной коммерции Groupon, браузер Mozilla Firefox.

Примечательным также является возможность разработки веб-приложения на платформе Node в облачной службе Microsoft Azure, несмотря на наличие собственного стека разработки веб-приложений WISA, использующего только продукты компании Microsoft.

Заключение

Каждая из технологий Node и PHP имеет свои достоинства и недостатки. Поэтому выбор технологии существенным образом зависит от разрабатываемого приложения.

Интегральная цифровая среда предоставления транспортных услуг имеет следующие особенности:

- необходимость обеспечения масштабируемости к изменению количества пользователей в системе;
- частый обмен информацией между клиентом и сервером в процессе поиска различных вариантов маршрутов;
- выполнение функций доступа к данным и функций расчета, требующих большого времени, наряду с запросами, время выполнения которых невелико;
- требования к скорости обработки информации на стороне сервера;
- выполнение параллельных алгоритмов обработки информации на стороне сервера и, как следствие, асинхронного выполнения программ;
- использование NoSQL базы данных в виду большого объема данных.

Учитывая перечисленные особенности, можно сделать вывод о предпочтительном использовании серверной платформы Node и стека MEAN (или его модификации в части базы данных) для реализации интегральной цифровой среды предоставления транспортных услуг.

Список литературы

1. Браун И. Веб-разработка с применением Node и Express. Полноценное использование стека JavaScript. – СПб.: Питер, 2017. – 336 с.
2. Клейменов А.А. Проблемы и перспективы развития процессного подхода в области «Цифровой железной дороги». – М.: Modern Science. Научно-информационный издательский центр «Институт стратегических исследований». – 2019. – № 10-2. – С. 92–96.
3. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. – СПб.: Питер, 2019. – 816 с.
4. Редмонд Э., Уилсон Джим Р. Семь баз данных за семь недель. Введение в современные базы данных и идеологию NoSQL. – М.: ДМК Пресс, 2018. – 384 с.
5. Рябова В. Утвержден план перехода РЖД на «цифровую железную дорогу» [Электронный ресурс] // D-russia.ru. – 2019. – URL: <http://www.d-russia.ru/utverzhdyon-plan-perehoda-rzhd-na-tsifrovuyu-zheleznuyu-dorogu.html> (дата обращения: 09.01.2020).
6. Buckler Craig. PHP vs Node.js [Электронный ресурс] // Sitepoint.com. – URL: <https://www.sitepoint.com/sitepoint-smackdown-php-vs-node-js> (дата обращения: 09.01.2020).
7. Louridas Panos. Component Stacks for Enterprise Applications // IEEE Software. – 2016. – Vol. 33, is. 2. – P. 93–98.
8. MEAN.JS [Электронный ресурс]. – URL: <http://www.meanjs.org> (дата обращения: 09.01.2020).
9. NODE [Электронный ресурс]. – URL: <https://www.nodejs.org> (дата обращения: 09.01.2020).
10. PHP [Электронный ресурс]. – URL: <https://www.php.net> (дата обращения: 09.01.2020).
11. PHP versus Node js fastest programs [Электронный ресурс] // Benchmarksgame-team.pages.net. – URL: <https://www.benchmarksgame-team.pages.debian.net/benchmarksgame/fastest/php-node.html> (дата обращения: 09.01.2020).
12. W3Techs [Электронный ресурс]. – URL: <https://www.w3techs.com> (дата обращения: 09.01.2020).

References

1. Braun I. Veb-razrabotka s primeneniem Node i Express. Polnocennoe ispol'zovanie stecka JavaScript. – SPb.: Piter, 2017. – 336 s.
2. Klejmenov A.A. Problemy i perspektivy razvitiya processnogo podhoda v oblasti «Cifrovoj zheleznoj dorogi». – M.: Modern Science. Nauchno-informacionnyj izdatel'skij centr «Institut strategicheskikh issledovanij». – 2019. – № 10-2. – S. 92–96.
3. Nikson R. Sozdaem dinamicheskie veb-sajty s pomoshch'yu PHP, MySQL, JavaScript, CSS i HTML5. – SPb.: Piter, 2019. – 816 s.
4. Redmond E., Uilson Dzhim R. Sem' baz dannyh za sem' nedel'. Vvedenie v sovremennye bazy dannyh i ideologiyu NoSQL. – M.: DMK Press, 2018. – 384 s.
5. Ryabova V. Utverzhdjen plan perekhoda RZHD na «cifrovuyu zheleznuyu dorogu» [Elektronnyj resurs] // D-russia.ru. – 2019. – URL: <http://www.d-russia.ru/utverzhdyon-plan-perehoda-rzhd-na-tsifrovuyu-zheleznuyu-dorogu.html> (data obrashcheniya: 09.01.2020).
6. Buckler Craig. PHP vs Node.js [Elektronnyj resurs] // Sitepoint.com. – URL: <https://www.sitepoint.com/sitepoint-smackdown-php-vs-node-js> (data obrashcheniya: 09.01.2020).
7. Louridas Panos. Component Stacks for Enterprise Applications // IEEE Software. – 2016. – Vol. 33, is. 2. – P. 93–98.
8. MEAN.JS [Elektronnyj resurs]. – URL: <http://www.meanjs.org> (data obrashcheniya: 09.01.2020).
9. NODE [Elektronnyj resurs]. – URL: <https://www.nodejs.org> (data obrashcheniya: 09.01.2020).
10. PHP [Elektronnyj resurs]. – URL: <https://www.php.net> (data obrashcheniya: 09.01.2020).
11. PHP versus Node js fastest programs [Elektronnyj resurs] // Benchmarksgame-team.pages.net. – URL: <https://www.benchmarksgame-team.pages.debian.net/benchmarksgame/fastest/php-node.html> (data obrashcheniya: 09.01.2020).
12. W3Techs [Elektronnyj resurs]. – URL: <https://www.w3techs.com> (data obrashcheniya: 09.01.2020).