

ПЕДАГОГИКА ВЫСШЕЙ ПРОФЕССИОНАЛЬНОЙ ШКОЛЫ

Государев Илья Борисович

канд. пед. наук, доцент

ФГБОУ ВПО «Российский государственный
педагогический университет им. А.И. Герцена»

г. Санкт-Петербург

ЭЛЕКТРОННОЕ ОБУЧЕНИЕ ВЕБ-ПРОГРАММИРОВАНИЮ НА ОСНОВЕ ТЕХНОЛОГИЙ ТЕСТИРОВАНИЯ JAVASCRIPT-СЦЕНАРИЕВ

Аннотация: статья содержит описание примера реализации электронного обучения разработке клиентских Javascript-сценариев с помощью технологий автоматизированного тестирования. Автор отмечает, что автоматизация тестирования ускоряет обратную связь и позволяет студентам быстрее и эффективнее корректировать код.

Ключевые слова: электронное обучение, JavaScript, юнит-тесты, модульное тестирование, веб-программирование, веб-разработка, PhantomJS.

Обучение веб-разработке в РГПУ им. А.И. Герцена осуществляется автором данной статьи и его коллегами в рамках бакалавриата по направлению «230100.62 – информатика и вычислительная техника» (профиль: «технологии разработки программного обеспечения») на дисциплинах модуля «Проектирование и разработка web-решений». В значительной степени обучение реализуется в электронной форме как вообще на кафедре [1], так и применительно к данному бакалавриату [2], однако спецификой содержания дисциплин и характером формируемых у студентов компетенций порождаются некоторые сложности реализации электронной формы обучения.

При стандартной разработке веб-решений в коммерческих организациях (например, в рамках Agile + Scrum) широко используются юнит-тесты, структура

которых адекватна структуре задействуемых паттернов проектирования и «вписана» в производственный процесс. Иными словами, разработчик находится в экосистеме, к которой «подключены» среда разработки, система контроля версий и другие инструменты, включая фреймворк тестирования (JsTestDriver, Karma, MochaJS и др.).

В ситуации обучения веб-проектированию и разработке веб-решений по целому ряду причин развернуть и поддерживать такую экосистему сложно. В частности, студентов 2 курса затруднительно обучать параллельно как тестированию, так и разработке сценариев «под тесты» (или тестов «под сценарии»), и нормальное для фирмы самотестирование заменяется контролем со стороны преподавателя, вынужденного лично проверять сотни решений.

В описываемой ситуации в качестве входного условия принимается тот факт, что студенты размещают свои решения (т. е. веб-страницы с JavaScript-кодом, решающим ту или иную задачу) на веб-серверах. В начале обучения они используют авторскую систему кодактор.рф (kodaktor.ru), где каждое небольшое решение занимает отдельную «доску» или «борд». Поэтому необходима технология тестирования, позволяющая удаленно обращаться к таким страницам по их адресам, вызывая сценарии с заранее сформированным набором входных данных. При этом ожидается, что результаты будут соответствовать эталонным результатам, вычисленным с помощью решения-образца (бенчмарк-кода).

В качестве такой технологии автором был выбран headless-движок Phantom.js (phantomjs.org) с открытым исходным кодом, позволяющий запускать на любом компьютере сценарии, имитирующие работу веб-браузера и сервера одновременно. На основе этой технологии была разработана прототипная схема процесса тестирования, сводящаяся к следующему:

1. Студенты реализуют веб-решения в виде сценариев с минимальной шаблонной частью. В частности, сценарий должен выдавать некий ответ в виде сериализованного JSON-объекта и записывать его в заголовок веб-страницы (`document.title = JSON.stringify(...)`).

2. Каждый раз при формулировании задания преподаватель предусматривает для него «кейс проверки», т. е. пару (множество наборов входных параметров сценария; множество ожидаемых ответов). Как только от студента поступает сообщение о готовности решения (т. е. фактически его URL) в сервисе сообщений LMS, преподаватель запускает сценарий проверки этого URL с этим кейсом в качестве параметра.

Например, если сценарий должен решать квадратное уравнение по трем коэффициентам, то кейс `http://kodaktor.ru/j/testing_86845` (`{"url":"http://kodaktor.ru/g/testing_583d4"}`, `«urlinputs»:[«?a=1&b=-2&c=-15», «?a=1&b=1&c=-12»,»?a=4&b=27&c=-7»]`, `"outputs»:[{"x1»:5,"x2»: -3}, {"x1»:3,"x2»: -4}, {"x1»:0.25,"x2»: -7}]`) позволяет отправить на адрес сценария три запроса с тремя наборами входных параметров и сравнить полученные ответы с эталонными: *phantom test1.js test1.case.json*

Вызов: `http://kodaktor.ru/g/testing_583d4?a=4&b=27&c=-7`

Получено: `{"x1»:0.25,"x2»: -7}`

Ожидалось: `{"x1»:0.25,"x2»: -7}`

Технология Phantom.js позволяет создавать асинхронные запросы к веб-страницам с помощью модуля *page*: `page.open(url, function(status) {..})`.

Разработанный автором статьи сценарий отправки запросов реализует перебор множества входных наборов кейса, при котором каждый последующий вызов веб-страницы осуществляется рекурсивно после завершения предыдущего асинхронного вызова в его callback-обработчике.

Автоматизация тестирования ускоряет обратную связь и позволяет студентам быстрее и эффективнее корректировать код. Соответствующий электронный учебно-методический комплекс представлен на сайте `kodaktor.ru`.

Список литературы

1. Аксютин П.А. Опыт построения среды электронного обучения и ее использование для преподавания дисциплины «Информационные технологии» // Электронное обучение в вузе и в школе: Материалы сетевой международной научно-практической конференции. – СПб.: Астерион, 2014. – С. 28–31.

2. Жуков Н.Н. Обучение программированию будущих инженеров направления «Информатика и вычислительная техника» // Электронное обучение в вузе и в школе: Материалы сетевой международной научно-практической конференции. – СПб.: Астерион, 2014. – С. 121–124.