


Выполнил(а) Батманов Даниил Евгеньевич, № группы P3107, оценка                       
Фамилия И.О. студента не заполнять

<b>Название статьи/главы книги/видеолекции</b> Автотесты на языке разметки или как мы в Lamoda тестируем микросервисы		
<b>ФИО автора статьи (или e-mail)</b> Кирилл Поляков @k_claim <a href="https://habr.com/ru/users/k_claim/">https://habr.com/ru/users/k_claim/</a>	<b>Дата публикации (не старше 2019 года)</b> 29.01.2021	<b>Размер статьи (от 400 слов)</b> 1995
<b>Прямая полная ссылка на источник или сокращённая ссылка (bit.ly, tr.im и т.п.)</b> <a href="https://habr.com/ru/company/lamoda/blog/539168/">https://habr.com/ru/company/lamoda/blog/539168/</a>		
<b>Теги, ключевые слова или словосочетания</b> Lamoda, Gonkey, язык разметки, Go, YAML, json, тестирование, микросервисы, решения.		
<b>Перечень фактов, упомянутых в статье</b> <ol style="list-style-type: none"> <li>1. Обычно, тесты состоят из трёх основных уровней: unit (тесты отдельных функций), integration (взаимодействие компонентов микросервиса и внешних систем), e2e (взаимодействие сервисов между собой).</li> <li>2. Для автотестов был выбран язык Go по следующим причинам: высокая поддержка от разработчиков, единый репозиторий размещения кода, запуск тестов вместе с кодом и, самое главное, возможность отладки.</li> <li>3. Автотест на Go, написанный на основе go testing не дал готового набора решений для компании, так как каждый автотест был бы совсем не универсальным.</li> <li>4. Для упрощения процесса написания тестов и стандартизации набора библиотек для тестирования разработчика Lamoda пришлось разработать свой инструмент: Gonkey.</li> <li>5. Структура для нового инструмента была задана в виде YAML.</li> <li>6. Принцип работы Gonkey: задание названия теста, определение используемого API-метода, выбор json для передачи, выбор json и кода ответа, ожидаемого для получения.</li> <li>7. За счёт использования в создании автотестов языка разметки YAML, Lamoda удалось создать инструмент для тестирования микросервисов, который в то же время решает поставленные задачи: упрощение процесса написания тестов и стандартизации наборов библиотек для тестирования.</li> </ol>		
<b>Позитивные следствия и/или достоинства описанной в статье технологии (минимум три пункта)</b> <ol style="list-style-type: none"> <li>1. Решение проблемы с переходом на Go (в отношении написания автотестов).</li> <li>2. Доступность инструмента: Gonkey находится в свободном доступе, соответственно, доступен каждому.</li> <li>3. Покрытие все микросервисов автотестами.</li> </ol>		
<b>Негативные следствия и/или недостатки описанной в статье технологии (минимум три пункта)</b> <ol style="list-style-type: none"> <li>1. Постоянная необходимость в доработке инструмента с развитием микросервисов.</li> <li>2. Несовершенство инструмента: Gonkey до сих пор находится в постоянном тесте, соответственно, постоянной доработке.</li> <li>3. Многие доработки находятся снаружи инструмента, это неудобно.</li> </ol>		
<b>Ваши замечания, пожелания преподавателю или анекдот о программистах<sup>1</sup></b> 		

<sup>1</sup> Наличие этой графы не влияет на оценку