

Тестирование веб-приложений

# Ответы на вопросы



# На этом уроке

1. Рассмотрим чек-лист тестирования веб-приложений.
2. Разберём вопросы собеседования.
3. Определим свой уровень компетентности по тестированию веб-приложений.

## Оглавление

### [Чек-лист тестирования веб-приложений](#)

[Функциональное тестирование](#)

[Интеграционное тестирование](#)

[Тестирование локализации и глобализации](#)

[Тестирование удобства использования](#)

[Кросс-платформенное тестирование](#)

### [Вопросы собеседования](#)

[Чем отличается PUT-запрос от POST?](#)

[Какие группы кодов состояний вы знаете?](#)

[Как будете тестировать таймаут?](#)

[Как будете проверять поп-ап, который появляется раз в сутки?](#)

[На тестирование передана новая страница «Профиль пользователя». Какие виды тестирования надо провести?](#)

[Техническое задание](#)

[Раздел описания кредитования](#)

[Где может быть баг?](#)

### [Практическое задание](#)

### [Используемые источники](#)

# Чек-лист тестирования веб-приложений

В этот чек-лист вошли только общие характеристики. Естественно, в тестируемом приложении может быть функциональность, для которой надо применять отдельный подход и создать отдельные сценарии. То же самое верно для производительности, удобства использования, безопасности и прочего тестирования, которое требуется приложению.

Чек-лист для тестирования веб-приложений состоит из шести разделов:

1. Функциональное тестирование.
2. Интеграционное тестирование.
3. Тестирование безопасности.
4. Тестирование локализации и глобализации.
5. Тестирование удобства использования.
6. Кросс-платформенное тестирование.

## Функциональное тестирование

В этом пункте важно убедиться, что наш продукт соответствует функциональной спецификации, упомянутой в документации по разработке.

### Что проверяем?

1. Тестирование форм:
  - a. Регистрация:
    - i. Пользователь с данными существует в системе.
    - ii. Пользователь с данными не существует в системе.
    - iii. Пользователь, заблокированный в системе, не может пройти повторную регистрацию.
  - b. Авторизация:
    - i. Пользователь с данными существует в системе.
    - ii. Пользователь существует в системе с введённым логином и паролем.
    - iii. Пользователь с введённым логином не существует в системе.
    - iv. Пользователь с введённым логином существует в системе, но пароль неверный.
    - v. Пользователь с введённым логином и паролем существует в системе, но заблокирован модерацией (страница заморожена).
    - vi. Валидация полей ввода.
  - c. Тестирование валидации всех обязательных полей:
    - i. Максимальная и минимальная длина.

- ii. Диапазон допустимых символов, спецсимволы.
  - iii. Обязательность к заполнению.
  - iv. Убедиться, что астериск (знак звёздочки) отображается у всех обязательных полей.
  - v. Убедиться, что система не отображает окно ошибки при незаполненных необязательных полях.
- d. Формы обратной связи.
  - e. Ссылки на пользовательские соглашения.

2. Поиск:

- a. Результаты существуют или не существуют.
- b. Корректное сообщение о пустом результате.
- c. Пустой поисковый запрос.
- d. Поиск по эмодзи.

3. Поля:

- a. Числовые поля: не принимают буквы, в этом случае отображается соответствующее сообщение об ошибке.
- b. Дробные значения, например, как система валидирует 1.1 и 1,1.
- c. Отрицательные значения в числовых полях, если они разрешены.
- d. Деление на ноль корректно обрабатывается.
- e. Тестирование максимальной длины каждого поля, чтобы убедиться, что данные не обрезаются или скрываются под многоточие.
- f. Тестирование всех полей ввода на спецсимволы.
- g. Проверка, что текст не выезжает за границы поля.

4. Всплывающие сообщения:

- a. Тестирование всплывающих сообщений: «Это поле ограничено N-знаками».
- b. Подтверждающие сообщения отображаются для операций обновления и удаления.
- c. Сообщения об ошибках ввода.

5. Фильтры:

- a. Протестировать функциональность сортировки: по возрастанию, убыванию, по новизне.
- b. Задать фильтры с выдачей.
- c. Задать фильтры, по которым нет выдачи.
- d. Фильтры по категориям или подкатегориям.
- e. Фильтры с радиусом поиска.

- f. Данные в выпадающих списках.
- 6. Функциональность доступных кнопок.
- 7. Наличие favicon.
- 8. Обработка различных ошибок: страница не найдена, тайм-аут, ошибка сервера и т. д.
- 9. Загруженные документы открываются правильно.
- 10. Пользователь может скачать, прикрепить или загрузить файлы или медиа — картинки, видео и т. д. Удалить эти файлы из вложений. Убедиться, что файлы уходят на сервер только после нажатия соответствующей кнопки.
- 11. Почтовая функциональность системы.
- 12. Кеш, cookie и сессии:
  - a. Пользователь очистил кеш браузера.
  - b. Посмотреть, что будет, если пользователь удалит куки, находясь на сайте.
  - c. Посмотреть, что будет, если пользователь удалит куки после посещения сайта.
- 13. DevTools:
  - a. Ошибки в Console.
  - b. Все стили загружаются.
  - c. Картинки загружаются.

## Интеграционное тестирование

Проводится, чтобы убедиться, что приложение совместимо со сторонними сервисами.

### Что проверяем?

- 1. Работа сторонних модулей: оплата, шаринг, карты.
- 2. Реклама: просмотр, переходы по рекламе, аналитика.
- 3. Метрика: переходы по страницам, показы элементов, клики.

### Тестирование безопасности

Такая проверка нацелена на поиск недостатков и пробелов, с точки зрения безопасности нашего приложения.

### Что проверяем?

- 1. Пользователь не может авторизоваться: под старым паролем, заблокирован в сервисе, достиг лимита авторизаций, ввёл чужой код верификации.

2. Страницы, содержащие важные данные, например, пароль, номер кредитной карты и CVC, ответы на секретные вопросы и прочее, открываются через HTTPS (SSL).
3. Пароль скрыт астерисками на страницах: регистрация, «забыли пароль», «смена пароля».
4. Корректное отображение сообщений об ошибках.
5. Завершение сессии после разлогина.
6. Доступ к закрытым разделам сайта.
7. SQL-инъекции.
8. Уязвимости Cross-Site Scripting (XSS).
9. HTML-инъекции.
10. Cookie хранятся в зашифрованном виде.
11. Роли пользователей и доступ к контенту.

## Тестирование локализации и глобализации

Тестирование интернационализации или глобализации веб-приложения включает тестирование приложения для различных местоположений, форматов дат, чисел и валют. А тестирование локализации включает тестирование веб-приложения с локализованными строками, изображениями и рабочими процессами для конкретного региона.

### Что проверяем?

1. Дата и время. Например, отображение времени, даты в соответствии с часовым поясом пользователя.
2. Смена языка и проверка перевода всех элементов веб-приложения исходя из выбранного языка.
3. Выбор номера телефона с разными кодами стран.
4. Определение местоположения пользователя и отображение соответствующего периметра ГЕО.
5. Отображение соответствующих символов валюты.

## Тестирование удобства использования

Это проверка навигации, контента, а также другая информация для пользователя.

### Что проверяем?

1. Отсутствие орфографических и грамматических ошибок, все страницы имеют корректные заголовки.

2. Выравнивание картинок, шрифтов, текстов.
3. Информативные ошибки, подсказки.
4. Подсказки есть для всех полей.
5. Отступы между полями, колонками, рядами и сообщениями об ошибках.
6. Кнопки имеют стандартный размер, цвет.
7. На сайте нет битых ссылок и изображений.
8. Неактивные поля отображаются серым цветом.
9. Поведение сайта при разных разрешениях экрана.
10. Скролл должен появляться только тогда, когда требуется.
11. Отображение чекбоксов и радиокнопок. Важно, чтобы кнопки были доступны с клавиатуры, и пользователь должен быть в состоянии пользоваться сайтом, используя только клавиатуру.
12. Отображение выпадающих списков.
13. Длинный текст скрывается под многоточие.
14. Корректный выбор даты.
15. Плейсхолдеры в полях.
16. Логотип ведёт на главную страницу сайта.
17. Переходы и навигация между страницами и разделами меню.

## Кросс-платформенное тестирование

Проводится, чтобы убедиться, что приложение совместимо с другими браузерами, различными оболочками, аппаратным обеспечением устройства.

### Что проверяем?

1. Тестирование в различных браузерах, например, Firefox, Chrome, Safari — это минимальный набор: анимация, вёрстка, шрифты, уведомления и т. д.
2. Тестирование в различных версиях ОС: Windows, Mac, Linux.
3. JavaScript-код работает в разных браузерах.
4. Просмотр на мобильных устройствах.

## Вопросы собеседования

### Чем отличается PUT-запрос от POST?

Например, наш сервис оперирует понятиями «новости» (news) и «запись» (post). Новостная страница может содержать множество записей.

Для добавления новой записи на страницу с идентификатором `id` будет использоваться метод POST с URL `mydomain/news/id/`. Сервис, ориентируясь на метод POST, сам присвоит подходящий идентификатор новости, добавит её в новостную ленту и вернёт URL созданной новости — для доступа к записи по GET или для удаления по DELETE.

Допустим, новость с идентификатором `post-id` уже создана и доступна по URL `mydomain/news/id/post/post-id`. Но владелец записи хочет её отредактировать. Для этого будет использоваться метод PUT с URL `mydomain/news/id/posts/post-id`, и передаваться обновлённая запись в теле запроса. Сервис, ориентируясь на метод PUT, удаляет старую новость и записывает новую, которая доступна по тому же URL.

**P.S.** Используется также метод POST, например, HTML 4 позволял использовать только методы GET и POST. Но всё же стоит придерживаться рекомендаций для единообразной трактовки методов всеми разработчиками.

## Какие группы кодов состояний вы знаете?

1. 1xx: Informational (информационные).
2. 2xx: Success (успешно).
3. 3xx: Redirection (перенаправление).
4. 4xx: Client Error (ошибка клиента).
5. 5xx: Server Error (ошибка сервера).

## Как будете тестировать таймаут?

В личном кабинете банка стоит ограничение времени бездействия — 10 минут. После этого сессия принудительно завершается, и отображается модальное окно: «Ваша сессия истекла. Авторизуйтесь повторно».

Надо проверить:

- отображается ли сообщение спустя 10 минут бездействия;
- не доступна ли пользователю информация из закрытой сессии;
- происходит ли переадресация на главную страницу после завершения сессии;
- не возобновляется ли предыдущая сессия после закрытия модального окна.

Тестирование сессии проводится в рамках тестирования безопасности.



## Как будете проверять поп-ап, который появляется раз в сутки?

На сайте есть поп-ап «Ежедневные бонусы», он показывается пользователю раз в день. Сайт создаёт файл cookie для пользователя со значением "showPopupDailyBonus":false, срок жизни которого, а значит и интервал повторного отображения поп-апа – сутки. По истечении этого срока напоминание для пользователя показывается вновь.

Задача: проверить, что спустя указанное время (24 часа) юзер увидит поп-ап. Можно подождать 1 день и проверить. А если этот поп-ап показывается раз в 30 дней?

Надо вручную изменить куки. Например, отредактировать значение ключа expires и задать срок действия куки — 1 минуту. Через минуту кука станет неактуальной, и пользователь снова увидит поп-ап с бонусами.

## На тестирование передана новая страница «Профиль пользователя». Какие виды тестирования надо провести?



← → ↻  ☆

 **LoDki**

Имя

Возраст

## Техническое задание

1. Создать страницу «Профиль пользователя». На форме должно быть два поля: «Имя» и «Возраст».

2. Поле «Имя» — текстовое поле. Максимальное количество символов — 15. Это поле должно состоять только из кириллических или латинских символов.
3. Поле «Возраст» — числовое поле. Максимальное значение символов — 2. Корректное значение в диапазоне 14–80.
4. На странице должна быть клиентская и серверная валидация.
5. При загрузке страницы кнопка «Сохранить» активна.

Надо провести функциональное тестирование полей:

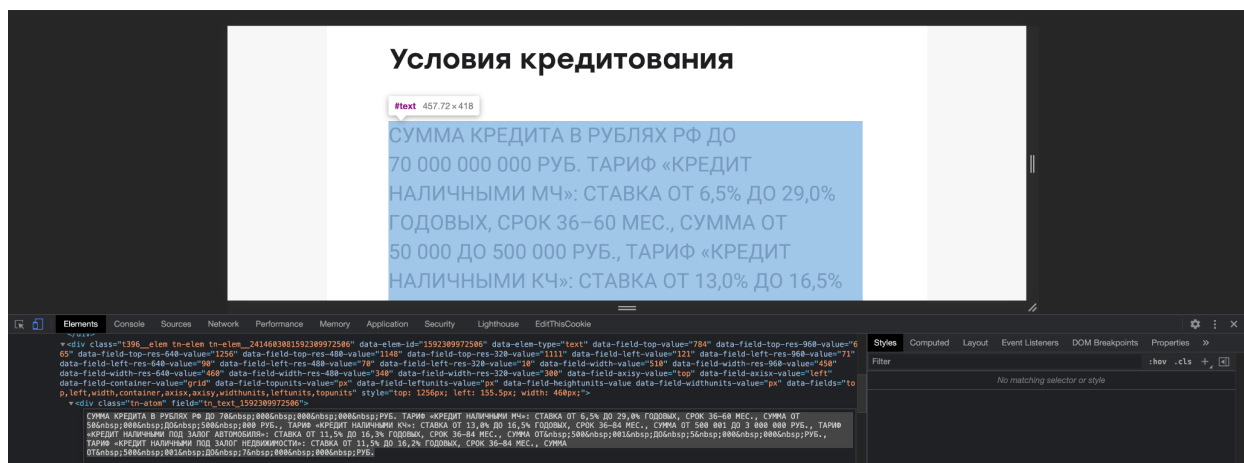
1. Максимальная и минимальная длина.
2. Диапазон допустимых символов, спецсимволы.
3. Обязательность к заполнению.
4. Убедиться, что астериск (знак звёздочки) отображается у всех обязательных полей.
5. Убедиться, что система не отображает окно ошибки при незаполненных необязательных полях.
6. Числовые поля: не принимают буквы, в этом случае отображается соответствующее сообщение об ошибке.
7. Дробные значения. Например, как система валидирует 1.1 и 1,1.
8. Отрицательные значения в числовых полях, если они разрешены.
9. Протестировать максимальную длину каждого поля.
10. Протестировать все поля ввода на спецсимволы.
11. Проверить, что текст не выезжает за границы поля.

Нефункциональное тестирование:

1. Соответствие макету.
2. Вычитка требований.

## Раздел описания кредитования

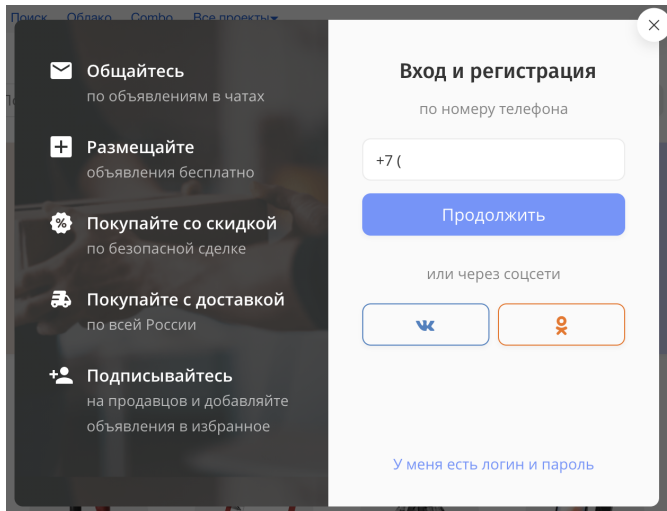
Допущена ошибка — сумма кредита не переносится на следующую строку.



Разработчик не добавил неразрывный пробел &nbsp;: «ДО 70&nbsp;000&nbsp;000&nbsp;000&nbsp;P».

## Где может быть баг?

Есть поле для ввода номера телефона. Отображается placeholder="+7 (\_\_\_\_) \_\_\_\_-\_\_-\_\_". При вводе номера телефона автоматически подставляется «+ 7», если первая введенная цифра — не «7», а пользователь вводит «7» или «8», то происходит автозамена на «+ 7».



Как вы думаете, где может быть баг?

После того как пользователь нажал «Продолжить», выполняется запрос:

```
https://somedomain.ru/web-api/auth/request_code
```

В теле запроса передаются параметры:

```
{
  phone: "9990752391",
  code_length: 6,
  use_callui: false
}
```

## Практическое задание

### Задание 1

Пройдитесь по методичкам предыдущих лекций курса «Тестирование веб-приложений» и выпишите списком все технологии, инструменты и навыки, которые вы изучили и могли бы добавить в своё резюме.

Формат сдачи: PDF файл со списком.

### **Задание 2 (бонусное)**

Напротив каждого пункта в списке поставьте оценку от 1 до 10, насколько вы поняли и сможете ответить на вопрос об этом пункте, если спросят на собеседовании.

Пример:

1. Postman — 6.
2. Chrome DevTools — 3.

Формат сдачи: просто дополните список из предыдущего задания и сохраните в PDF.

### **Задание 3 (бонусное)**

Расположите пункты в порядке убывания по степени важности по вашему мнению.

1. Самый верхний — тот, о котором, очевидно, спросят на собеседовании, так как он, скорее всего, очень важен в работе.
2. Самый нижний — тот, о котором никто не спросит, так как он редко применяется в работе.

Формат сдачи: просто дополнить список из предыдущего задания и также сохранить в PDF.

### **Задание 4 (бонусное)**

Подготовьте резюме:

- если вы в СНГ, то на hh.ru;
- если в другой стране — в PDF или на linkedin.com.

Формат сдачи: ссылка на резюме в hh.ru, linkedin.com или на google drive, где лежит ваше резюме в pdf.

Обязательно проверьте, открывается ли ваше резюме в режиме инкогнито в Chrome.

## **Используемые источники**

1. [Чек-лист тестирования веб-приложений.](#)