

## Тестовое задание на позицию Junior PHP Developer

В этом тестовом задании Вам нужно разработать онлайн каталог сотрудников для компании с более чем 50,000 сотрудников.

### Общая информация

1. Тестовое задание разделено на две части. Часть №1 обязательная и мы примем Вашу кандидатуру к рассмотрению только в том случае если часть №1 выполнена. Часть №2 опциональная, по количеству задач из этой части которые Вы смогли выполнить мы сможем оценить Ваш уровень опыта и навыков в области веб разработки. Если у Вас уже есть опыт работы с фреймворками Laravel и/или Symfony – попробуйте выполнить все или как можно больше задач из части №2, это повысит ваши шансы по отношению к другим кандидатам.
2. Вы должны отправить Ваше решение для тестового задания на нашу электронную почту [hr@abz.agency](mailto:hr@abz.agency) **в виде ссылки на github/bitbucket репозиторий**. Если Вы пришлете Ваше решение в любом другом виде (в виде ссылки на zip архив, прикрепите zip архив к письму, и др.) – Ваша кандидатура не будет нами рассмотрена!
3. Вам необходимо также приложить ссылкой или вложением подробное резюме, контактный номер телефона и ник скайпа.
4. Укажите в письме список выполненных пунктов для каждой части, выбранный вами фреймворк. Если Вы сделали не все пункты тестового задания – пожалуйста, укажите причину по которой вы их не выполнили (не хватило времени, не хватает опыта/знания, что-то ещё).

### Технические требования

- Laravel 5.4 / Symfony 2.8 LTS / Symfony 3.x
- MySQL 5.6+
- PHP 5.6 / 7.x

При выполнении тестового задания Вы можете дополнительно использовать любые сторонние PHP и/или Javascript/CSS библиотеки, без всяких ограничений. Все 3rd party PHP/Javascript/CSS библиотеки должны быть добавлены в проект через composer/bower/npm/yarn если библиотека поддерживает такой способ установки. У нас нет никаких требований к оформлению фронтенд части, но аккуратный вид приветствуется и добавим вам бонусных пунктов.

### Часть №1 (обязательная)

Создайте веб страницу, которая будет выводить иерархию сотрудников в древовидной форме.

- Информация о каждом сотруднике должна храниться в базе данных и содержать следующие данные:
  - ФИО;
  - Должность;
  - Дата приема на работу;
  - Размер заработной платы;
- У каждого сотрудника есть 1 начальник;
- База данных должна содержать не менее 50 000 сотрудников и 5 уровней иерархий.
- Не забудьте отобразить должность сотрудника.

### Часть №2 (опциональная)

1. Создайте базу данных используя миграции Laravel / Symfony.
2. Используйте Laravel / Symfony seeder для заполнения базы данных.
3. Используйте Twitter Bootstrap для создания базовых стилей Вашей страницы.
4. Создайте еще одну страницу и выведите на ней список сотрудников со всей имеющейся о сотруднике информацией из базы данных и возможностью сортировать по любому полю.

5. Добавьте возможность поиска сотрудников по любому полю для страницы созданной в задаче 4.
6. Добавьте возможность сортировать (и искать если Вы выполнили задачу №5) по любому полю без перезагрузки страницы, например используя ajax.
7. Используя стандартные функции Laravel / Symfony, осуществите аутентификацию пользователя для раздела веб сайта доступного только для зарегистрированных пользователей.
8. Перенесите функционал разработанный в задачах 4, 5 и 6 (используя ajax запросы) в раздел доступный только для зарегистрированных пользователей.
9. В разделе доступном только для зарегистрированных пользователей, реализуйте остальные CRUD операции для записей сотрудников. Пожалуйста заметьте, что все поля касающиеся пользователей должны быть редактируемыми, включая начальника каждого сотрудника.
10. Осуществите возможность загружать фотографию сотрудника и отобразите ее на странице, где можно редактировать данные о сотруднике. Добавьте дополнительную колонку с уменьшенной фотографией сотрудника на странице списка всех сотрудников.
11. Осуществите возможность перераспределения сотрудников в случае изменения начальника (бонусом может быть то, что вы сможете это осуществить с применением встроенных механизмов/парадигм, предлагаемых Laravel / Symfony ORM).
12. Реализуйте ленивую загрузку для дерева сотрудников. Например, показывайте первые два уровня иерархии по умолчанию и подгружайте 2 следующих уровня или всю ветку дерева при клике на сотрудника второго уровня.
13. Реализуйте возможность менять начальника сотрудника используя drag-n-drop сразу в дереве сотрудников.
14. Создайте структуру базы данных используя MySQL Workbench (не забудьте закоммитить проект MySQL Workbench в git) и сгенерируйте файл(ы) миграций с помощью Laravel / Symfony из существующей БД MySQL, или прямо из файла проекта MySQL Workbench.

Пожалуйста не забудьте, что Ваше тестовое задание должно быть предоставлено в виде ссылки на github/bitbucket репозиторий. Мы рассмотрим каждое задание соответствующее заявленным выше требованиям. Рассмотрение занимает некоторое время, поэтому не стоит перезванивать нам на следующий день, чтобы узнать результат. Так же не стоит спешить и отправлять только часть выполненных заданий, если чувствуете что можете сделать больше. У нас открыто несколько вакансий и мы всегда в поиске талантов, проявите себя! Мы желаем Вам удачи и с нетерпением ждем Ваше выполненное задание.